ÍNDICE GENERAL - MANUALES

CAPITULO 1 1

1 INGRESO AL SISTEMA 1

1.1 Autentificación de Usuario 1

1.2 Error de Autentificación 2

1.3 Clave no valida 2

1.4 Usuario no Registrado 3

1.5 Clave de usuario errada 3

CAPITULO 2 5

2 OPCIONES DEL MENÚ PRINCIPAL DEL ADMINISTRADOR DE SISTEMA 5

2.1 Menú Principal 5

2.1.1 Mantenimiento de Usuarios 5

2.1.1.1 Listar Todos los Usuarios del Sistema. 6

2.1.1.2 Listar Usuarios por Estado. 7

2.1.1.3 Listar Usuarios por Nombre 7

2.1.1.4 Listar por Usuario de Ingreso 8

2.1.1.5 Creación de un Usuario 8

2.1.1.6 Asignar un Rol a un Usuario 9

2.1.2 Mantenimiento de Roles 11

2.1.2.1 Listar todos los Roles 12

2.1.2.2 Listar de Roles por el Estado. 12

2.1.2.4 Listar Roles por la Descripción 13

2.1.2.5 Creación de un Rol 14

2.1.2.6 Asignar un Menú a un Rol 15

2.1.3 Mantenimiento de Menús 17

2.1.3.2 Listar los menús por Nombre 18

2.3.1.3 Listar menús por url requerido 19

2.1.3.4 Listar por menús mostrables o no 19

2.3.1.5 Listar menús por estado 20

2.3.1.6 Listar los menús por agrupados o no agrupados. 21

2.3.1.7 Creación de un nuevo Menú 21

2.3.1.8 Creación de un Submenú 23

CAPITULO 3 26

3 MANUAL TECNICO 26

MANUAL DE USUARIO

CAPÍTULO 1

1.- INGRESO AL SISTEMA ACADEMICO

1.1 Autentificación de Usuario

En esta pantalla procedemos a ingresar el usuario ya sea este coordinador, profesor, alumno o administrador del sistema, así mismo se ingresara la clave para cada uno de los usuarios que tienen acceso al sistema.

Si fuese el caso de que el usuario no este registrado, este no podrá tener acceso al sistema académico.

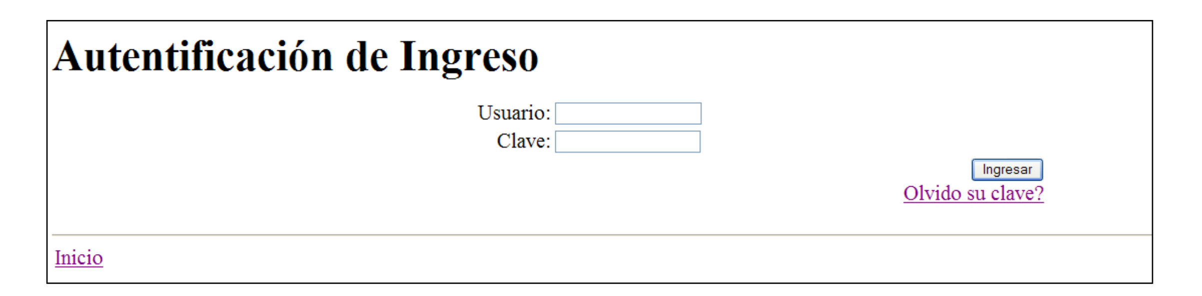


Fig. 1.1 Pantalla de Ingreso de Usuario.

1.2 Error de Autentificación

Cabe mencionar que si la clave es obviada en el ingreso nos aparecerá un mensaje el cual indica que la clave es obligatoria.



Fig. 1.2 Error de Autentificación.

1.3 Clave no valida

En esta ventana podemos observar que si cualquiera de los dos parámetros necesarios de ingreso esta erróneo, el sistema me arroja un mensaje indicando el porque se esta denegando el acceso, tal como se muestra en la fig. 1.3

Autentificación de Ingreso

Tu login no fue exitoso. Intenta de nuevo!
Razón: Clave no válida.

Usuario:
Clave:

Ingresar
Olvido su clave?

Fig. 1.3 Clave no valida

1.4 Usuarios No Registrado

En esta pantalla observamos como aparecerá un mensaje de error diciéndonos que el usuario no esta registrado en el Sistema por cuanto debemos intentar ingresar nuevamente o comunicarse con el administrador para verificar si esta registrado o activo el usuario.

Autentificación de Ingreso

Tu login no fue exitoso. Intenta de nuevo!
Razón: Usuario no registrado en el Sistema.

Usuario:
Clave:

Ingresar
Olvido su clave?

Fig. 1.4 Mensaje de usuario no registrado.

1.5 Clave de usuario errada

En caso de que la clave no sea valida y el usuario no la recuerde, tiene la opción de poder cambiarla presionando el link ***Olvido su clave?*** en la cual mostrara la siguiente pantalla

Autentificación de Ingreso

Tu login no fue exitoso. Intenta de nuevo!
Razón: Clave no válida.

Usuario:
Clave:

Ingresar
Olvido su clave?

Fig. 1.5 Clave de usuario errada

En esta pantalla, se pedirá al usuario que recuerde la respuesta de la pregunta secreta que ingreso inicialmente y debe presionar el botón grabar, como se muestra en la figura 1.6



Fig. 1.6 Olvido de clave

En ese momento le aparece una nueva pantalla donde debe ingresar la nueva clave y la confirmación de la misma, luego debe grabar el cambio de clave.



Fig. 1.7 Cambio de clave

Después de grabar la clave, el usuario debe autenticarse de nuevo en la pantalla principal del portal para acceder a sus respectivos menús y submenús de acuerdo a sus roles asignados.

CAPÍTULO 2

2.- OPCIONES DEL MENÚ PRINCIPAL DEL ADMINISTRADOR DE SISTEMA

2.1 Menú Principal

En el menú principal del Administrador se mostraran tres opciones:

- Mantenimiento de Usuarios
- Mantenimiento de Roles
- Mantenimiento de Menús

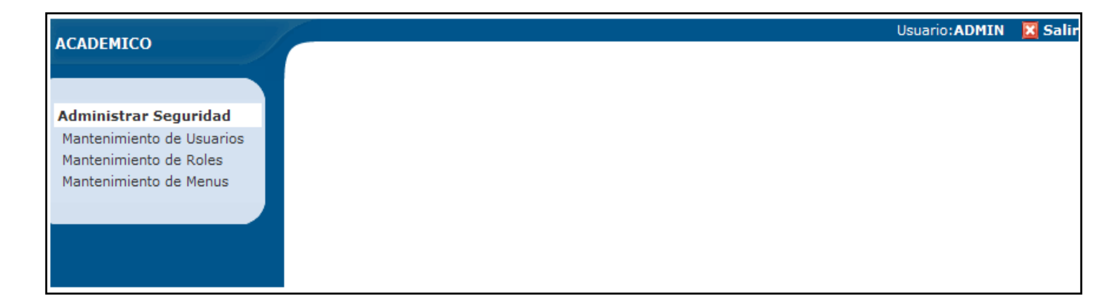


Fig. 2.1 Menú Inicial del Administrador

2.1.1 Mantenimiento de Usuarios

En esta pantalla nos muestra las diferentes formas en que podemos realizar las consultas y la creación de los usuarios, las cuales pueden ser las siguientes:

1. Listar todos los usuarios
2. Listar usuarios por el estado(A/I)

3. Listar usuarios por Nombre
4. Listar por Usuario de Ingreso
5. Creación de un Usuario
6. Asignación de un Rol a un Usuario

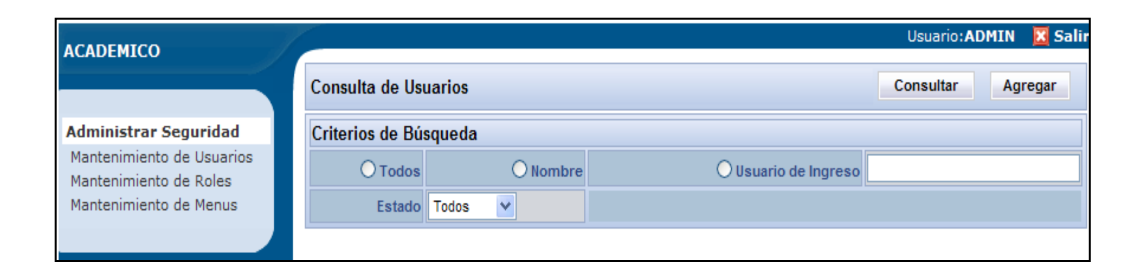
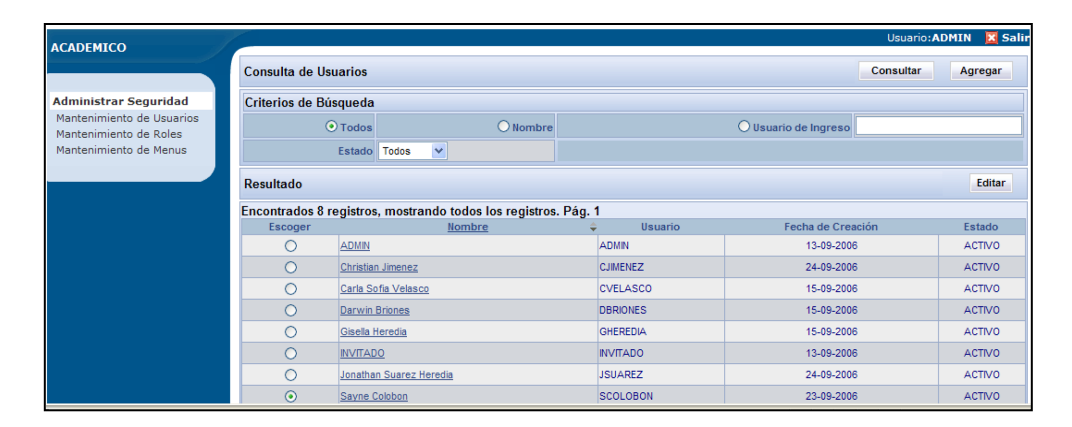


Fig. 2.2 Mantenimiento de Usuarios

2.1.1.1 Listar Todos los Usuarios del Sistema.

Esta consulta puede ser masiva sin filtros, listando todos los usuarios que se encuentran almacenados en el sistema.



Escoger	Nombre	Usuario	Fecha de Creación	Estado
	ADMIN	ADMIN	13-09-2006	ACTIVO
	Christian Jimenez	CJIMENEZ	24-09-2006	ACTIVO
	Carla Sofia Velasco	CVELASCO	15-09-2006	ACTIVO
	Darwin Briones	DBRIONES	15-09-2006	ACTIVO
	Gisella Heredia	GHEREDIA	15-09-2006	ACTIVO
	INVITADO	INVITADO	13-09-2006	ACTIVO
	Jonathan Suarez Heredia	JSUAREZ	24-09-2006	ACTIVO
	Sayne Colobon	SCOLOBON	23-09-2006	ACTIVO

Fig. 2.3 Lista de Todos los Usuarios

2.1.1.2 Listar Usuarios por Estado.

La consulta puede ser filtrada por el estado del usuario. Podemos hacer un filtro de Usuarios Activos, en donde me enlista todos los usuarios que pueden tener acceso al sistema a través de la página web o hacer un filtro de Usuarios Inactivos, el cual filtra todos los usuarios que por alguna razón han dejado de estudiar o han dado mal uso al sistema académico y por ende han sido deshabilitados.

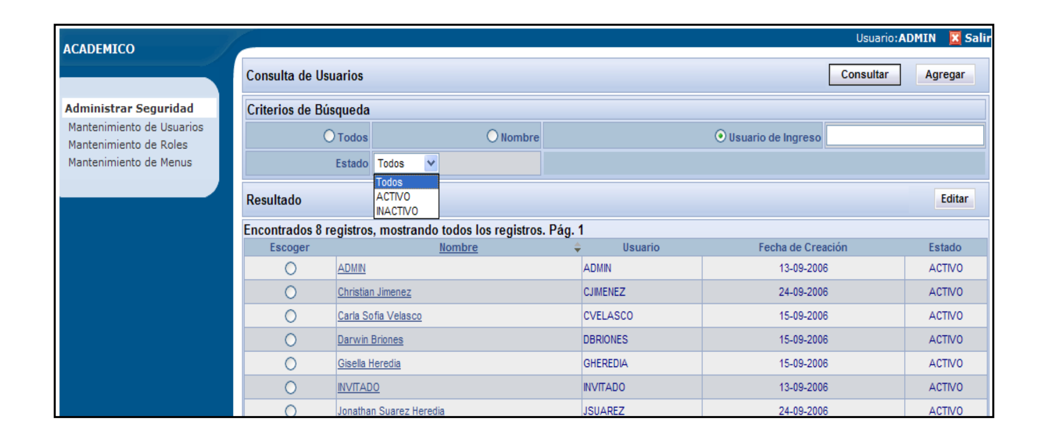


Fig. 2.4 Listar Usuarios por Estado

2.1.1.3 Listar Usuarios por Nombre

Se puede consultar en esta opción dando clic en Nombre y digitando el nombre y/o apellido del usuario que se desea obtener información.

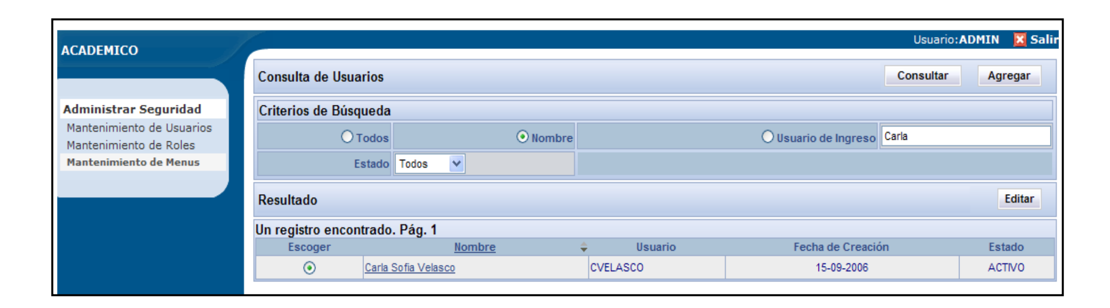


Fig. 2.5 Listar Usuarios por Nombre

2.1.1.4 Listar por Usuario de Ingreso

Se podrá también realizar la consulta especifica por usuario, escogemos la opción *usuario de ingreso* y digitamos el usuario (dbriones)



Fig. 2.6 Listar por Usuario de Ingreso

2.1.1.5 Creación de un Usuario

En la figura 2.2, al dar clic en el botón Agregar, nos aparece la siguiente pantalla en la cual nos pedirá ciertos datos necesarios del usuario para registrarlo en nuestra base.

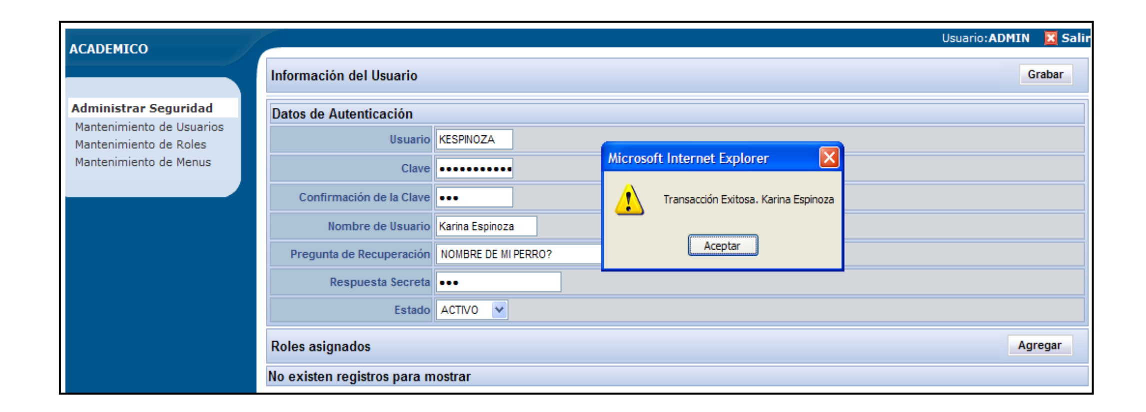


Fig. 2.7 Creación de un Usuario

Una vez que llenamos la información damos clic en el boton Grabar y nos muestra un mensaje que nos indica que la "Transacción fue exitosa"

2.1.1.6 Asignar un Rol a un Usuario

Para poder realizar esta asignación, primeramente debe estar creado el rol y asignado a este su respectivo menú y submenú si lo tuviere.

Se debe ir a la opción Mantenimiento de Usuario, elegir el usuario al cual se le va a asignar el rol, le aparecerá los datos del usuario



Fig. 2.8.1 Asignar un Rol a un Usuario

Para asignarle un Rol, en esta misma pantalla hay un botón "Agregar" al presionarlo le enlistara todos los roles posibles para la asignación.

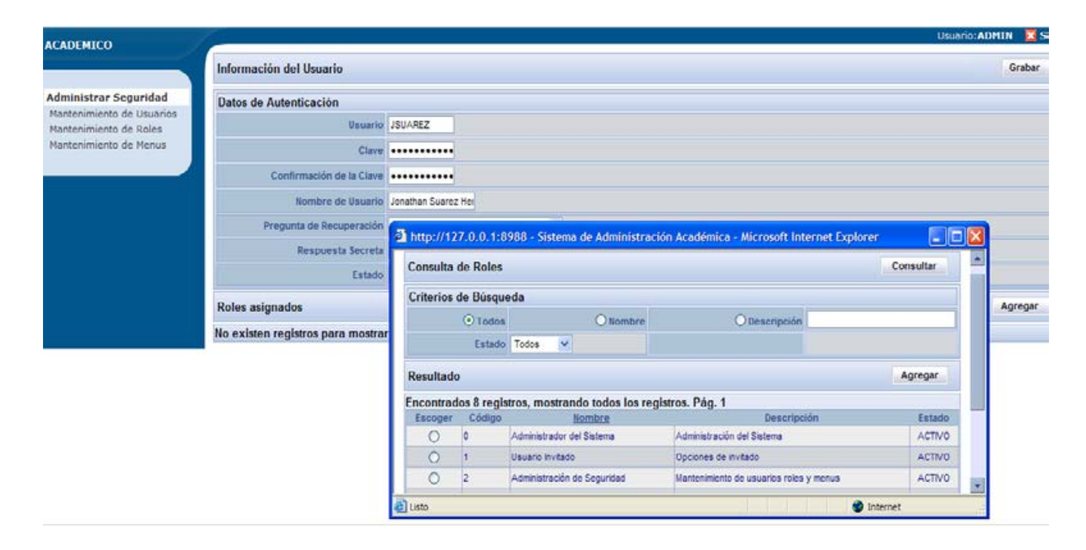


Fig. 2.8.2 Asignar un Rol a un Usuario

Se debe elegir el rol que se desee, haciendo click en el botón de la columna Escoger y luego presionar el botón “Agregar”. Después le aparecerá la siguiente pantalla, donde se debe presionar Aceptar y luego el botón grabar.

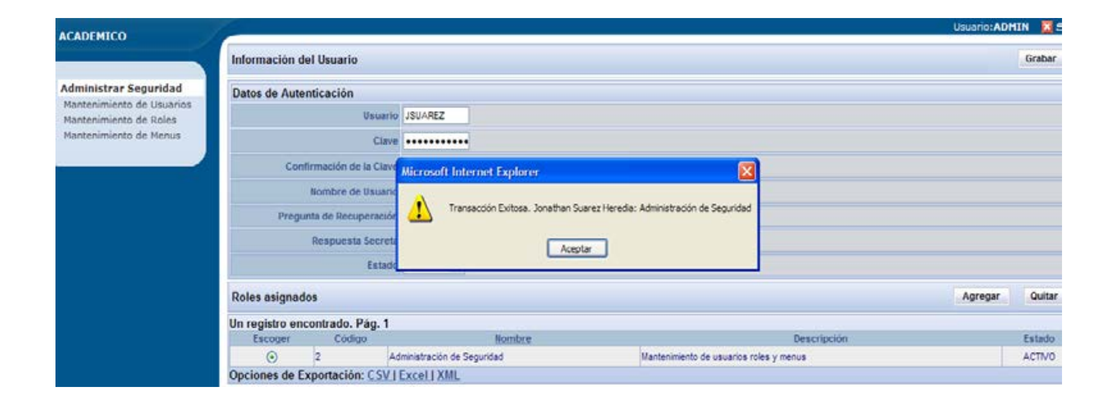


Fig. 2.8.3 Asignar un Rol a un Usuario

Donde le aparecerá una nueva pantalla indicando que la transacción se guardo correctamente.

Luego se presiona aceptar y después grabar donde le mostrara la siguiente pantalla donde indica que la transacción fue exitosa.

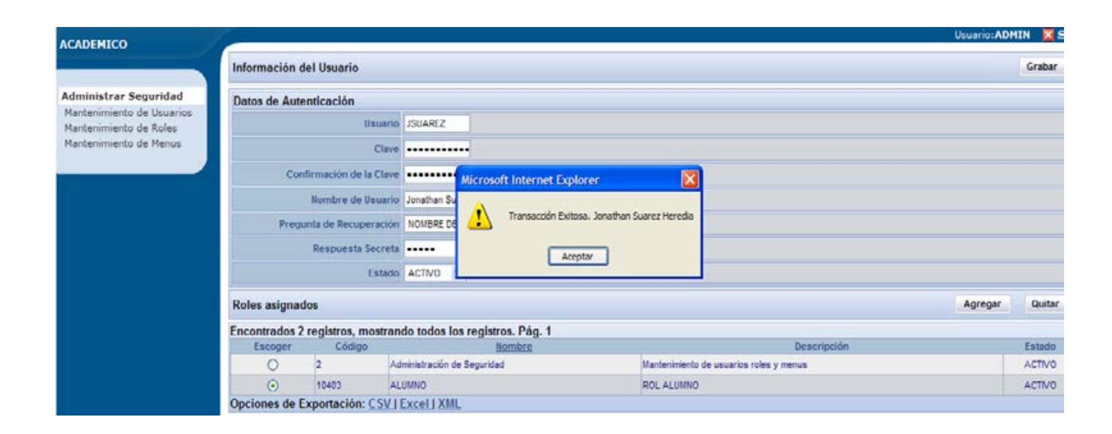


Fig. 2.8.4 Asignar un Rol a un Usuario

2.1.2 Mantenimiento de Roles

En esta pantalla se muestran las diversas opciones en que podemos realizar las consultas de roles, creaciones y asignaciones a los usuarios, los cuales pueden ser las siguientes:

1. Listar todos los roles
2. Listar roles por el estado(A/I)
3. Listar roles por Nombre
4. Listar roles por la Descripción
5. Creación de un Rol
6. Asignar un menú a un Rol

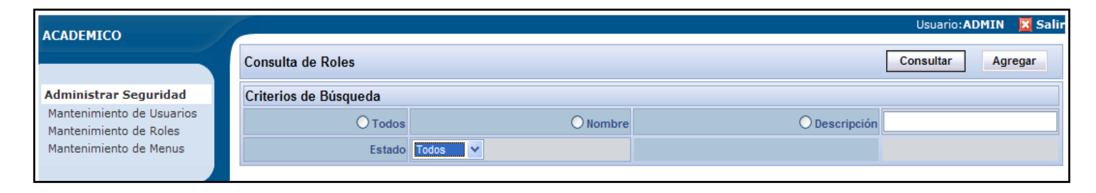


Fig. 2.9 Mantenimiento de Roles

2.1.2.1 Listar todos los Roles

Esta consulta puede ser masiva sin filtros, listando todos los roles que se encuentran almacenados en el sistema.



Fig. 2.10 Listar todos los Roles

2.1.2.2 Listar de Roles por el Estado.

Los roles también pueden ser consultados por su estado que puede ser:

Activo→ Un Rol activo es el que puede tener un usuario

Inactivo → Un Rol inactivo es el que se ha dejado de usar puede ser porque se elimino el cargo de la empresa, pero que está en el sistema definido en caso de ser requerido nuevamente.

Ambos (Todos)→ Muestra todos los roles, tanto los activos como los inactivos.

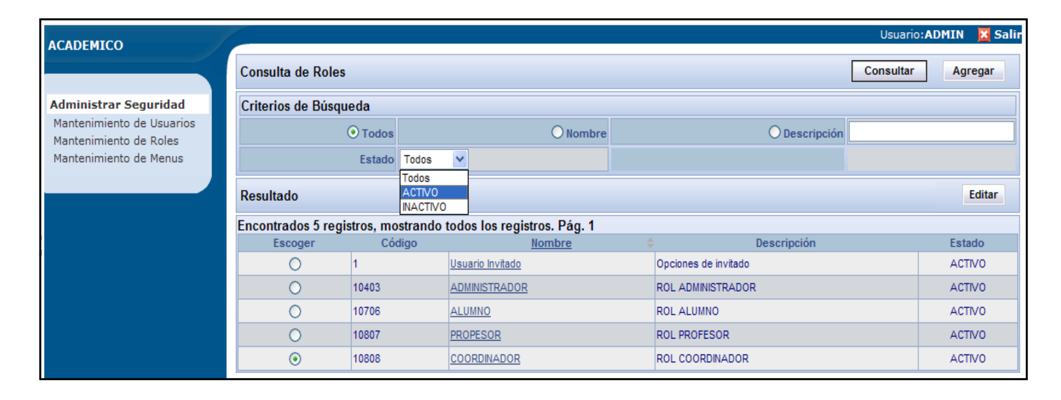


Fig. 2.11 Listar Roles por el Estado

Escogemos la opción *Nombre* y ponemos el nombre del rol que deseo buscar, tal como se muestra en la Fig. 2.12

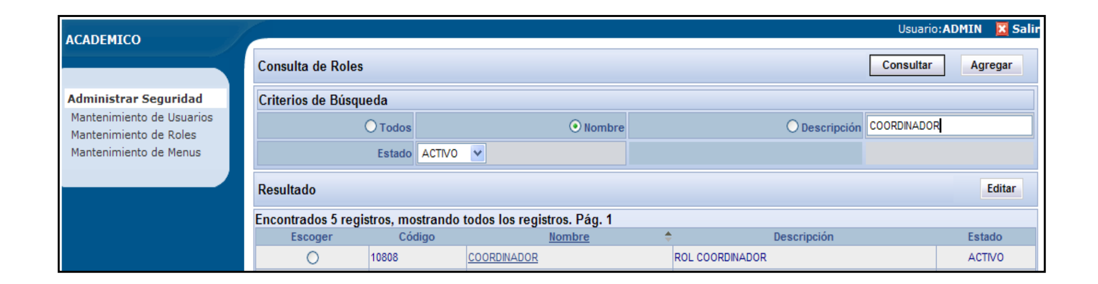


Fig. 2.12 Listar Roles por Nombre

2.1.2.4 Listar Roles por la Descripción

En esta consulta los roles pueden ser listados poniendo una descripción del rol o roles a consultar.

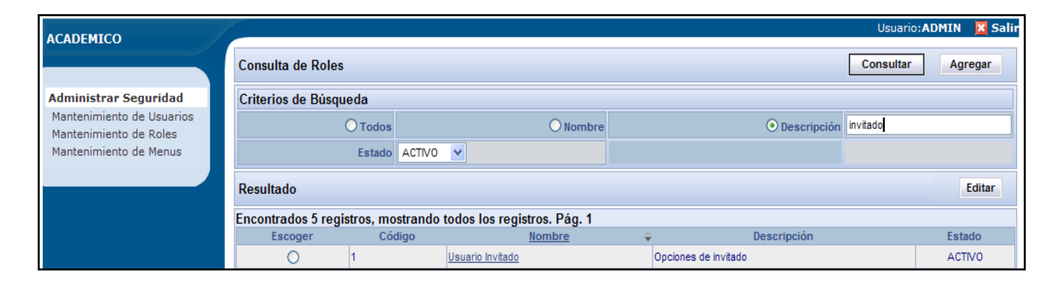


Fig. 2.13 Listar Roles por la Descripción

2.1.2.5 Creación de un Rol

En la figura 2.8, al dar clic en el botón Agregar, nos aparece la siguiente pantalla en la cual se debe ingresar el nombre del Rol, la descripción y el estado del mismo.

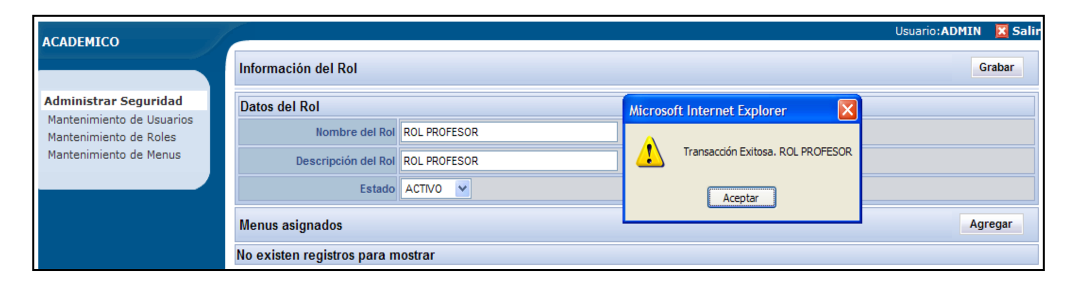


Fig. 2.14 Creación de un Rol

Enseguida se debe presionar el botón Grabar y aparecerá un mensaje indicando que la creación del Rol fue un éxito.

En este momento podemos hacer una consulta de roles y el rol creado recientemente podrá ser visualizado.

2.1.2.6 Asignar un Menú a un Rol

Para poder realizar esta asignación, primeramente debe estar creado el menú y su respectivo submenú; si este lo tuviere.

Se debe ir a la opción Mantenimiento de Rol, elegir el Rol al cual se le va a agregar el menú.

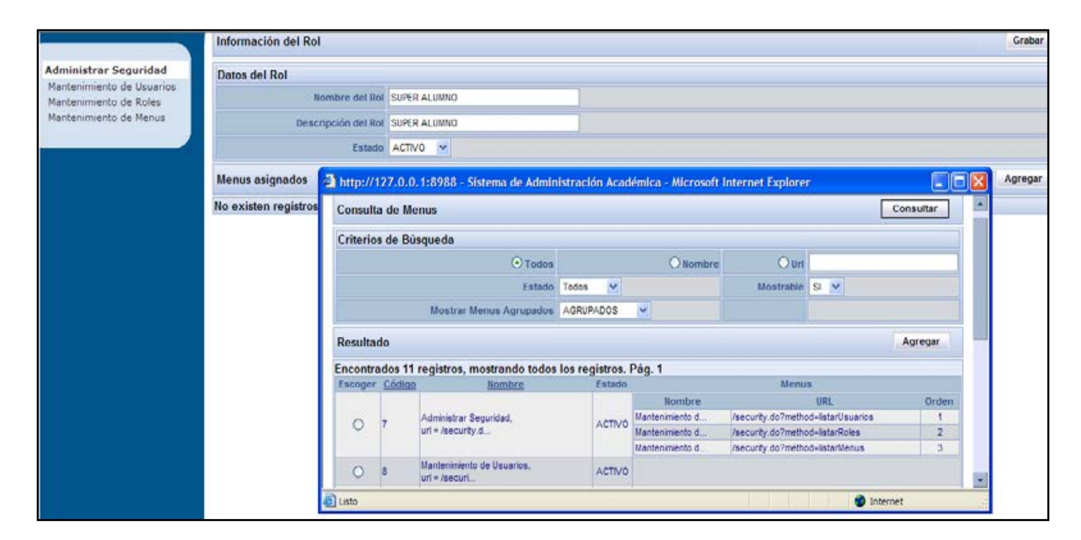


Fig. 2.15.1 Asignación de un menú a un Rol

En esta misma pantalla hay un botón “Agregar” al presionarlo le enlistara todos los menús posibles para la asignacion.

Se debe elegir el que se desee haciendo click en el boton de la columna Escoger y luego presionar el boton “Agregar”. Luego le aparecera la siguiente pantalla, donde se debe presionar Aceptar y luego el boton grabar .



Fig. 2.15.2 Asignación de un menú a un Rol

Donde le aparecerá una nueva pantalla indicando que la transacción se guardo correctamente.



Fig. 2.15.3 Asignación de un menú a un Rol

En caso que se requiera que dicho Rol posea solo una de las dos opciones que se posee el menú, se debe hacer un clic en el cuadrito de Quitar sobre la opción que no se requiera, en ese momento le aparecerá la siguiente pantalla

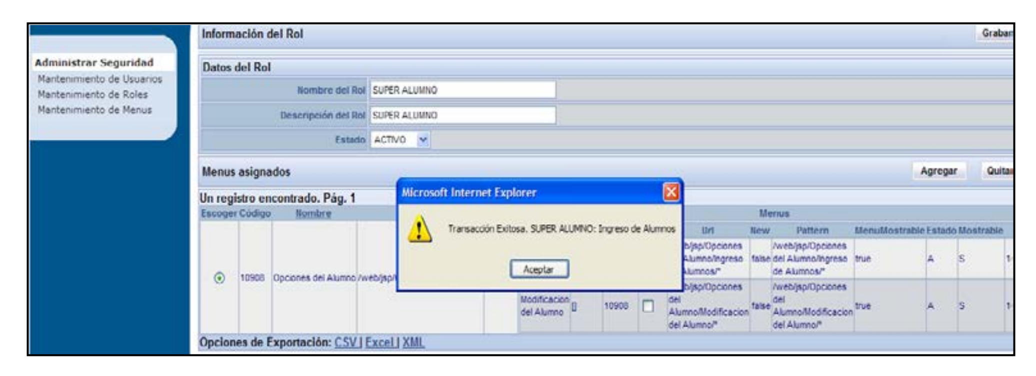


Fig. 2.15.4 Asignación de un menú a un Rol

Se debe presionar Aceptar y después presionar el botón “Grabar” donde le mostrara la siguiente pantalla que indica que la transacción fue exitosa. Se presiona Aceptar y luego Grabar.

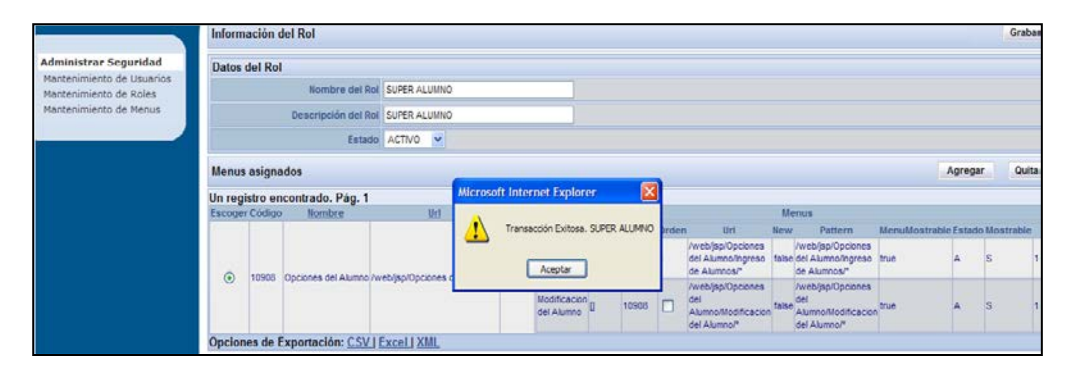


Fig. 2.15.5 Asignación de un menú a un Rol

2.1.3 Mantenimiento de Menús

Esta pantalla nos muestra las diferentes formas en que podemos consultar los menús del sistema y a su vez crear submenús, estas opciones pueden ser:

1. Listar todos los menús
2. Listar los menús por Nombre
3. Listar los menús por el Url correspondiente.
4. Listar menús Mostrables o no mostrables.
5. Listar menús por estado
6. Listar los menús por su agrupamiento
7. Creación de un Menú
8. Creación de Submenú

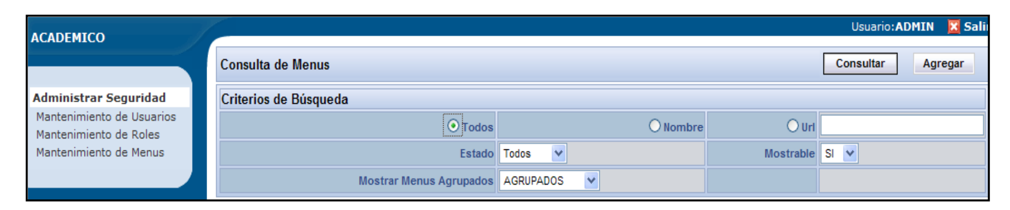


Fig. 2.16 Mantenimiento de Menús

Esta consulta puede ser masiva sin filtros, listando todos los menús que se encuentran almacenados en el sistema.



Fig. 2.17 Listar todos los Menús

2.1.3.2 Listar los menús por Nombre

Se puede consultar en esta opción dando clic en Nombre y digitando el nombre del menú que se desea obtener información.



Fig. 2.18 Listar los Menús por Nombre

2.3.1.3 Listar menús por url requerido

La consulta de los menús que se hace por la dirección URL que es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los menús de información disponibles; nos ayuda a identificar cada jsp que se va a ejecutar la opción requerida del menú a consultar.

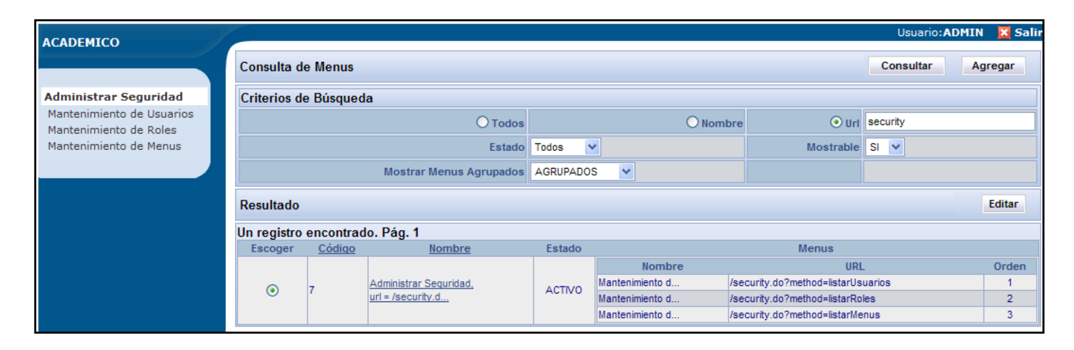


Fig. 2.19 Listar menús por url requerido

2.1.3.4 Listar por menús mostrables o no

En esta opción, si elegimos:

Mostrable SI, nos enlistara los menús que están asignados a un rol y que se muestran para los usuarios.

Mostrable NO, nos enlistara los menús que se están asignados a un rol y que no se muestran para los usuarios.

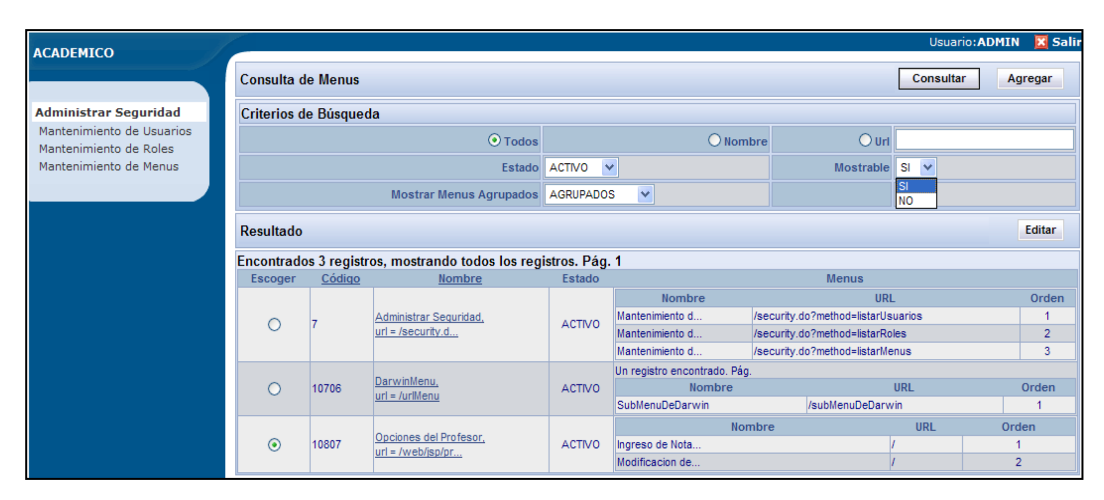


Fig. 2.20 Listar por menús mostrables o no

2.3.1.5 Listar menús por estado

En esta opción, la consulta puede ser filtrada por el estado del menú. Se puede hacer un filtro por:

Menús Activos, en donde se enlista todos los menús a los que se puede tener acceso en el sistema a través de la página web.

Menús Inactivos, el cual filtra todos los menús que por alguna razón se han desactivado ya sea porque no se usan o no se requiere.

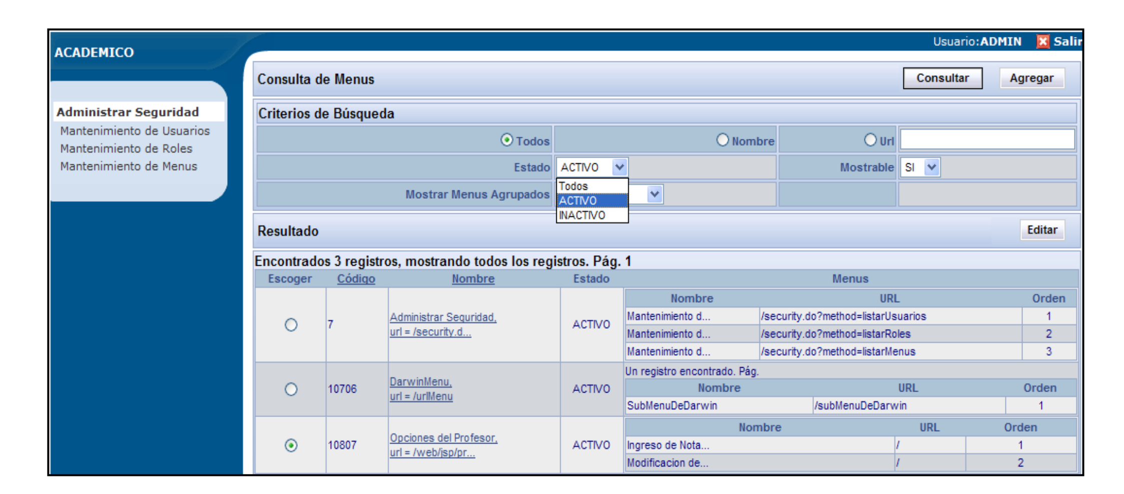


Fig. 2.21 Listar menus por estado

2.3.1.6 Listar los menús por agrupados o no agrupados.

La consulta de los menús agrupados, nos muestra los menús contenedores de otros menús a los que se les denomina submenús.

La consulta de los menús no agrupados, nos muestra cada uno de los menús y submenús que se han creado en el sistema, estén o no contenidos en otro.

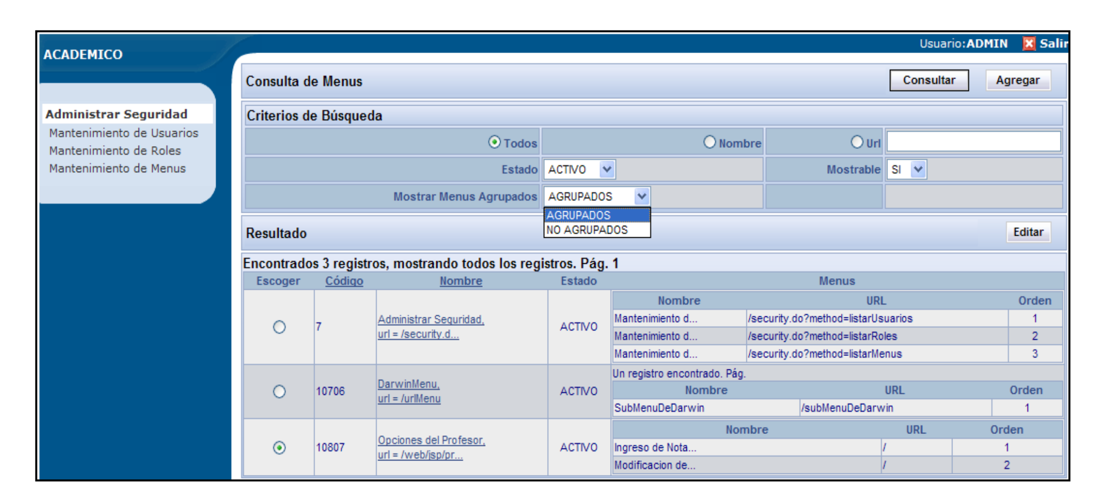


Fig. 2.22 Listar los menús por agrupados o no agrupados

2.3.1.7 Creación de un nuevo Menú

En consultar menú presionamos el botón agregar en donde nos aparece la ventana siguiente la cual nos permite ingresar el nombre del menú, el menú padre (*), url de acción o permiso, el orden o nivel, mostrabilidad y el estado de dicho menú**.**



Fig. 2.23.1 Creación de un nuevo Menú

(*) El menú padre se llena cuando estemos creando un submenú. En este caso estamos creando un menú padre este campo se deja vacío y el orden o nivel es 0 (cero). El url o permiso se debe llenar con los actions de validación.



Fig. 2.23.2 Creación de un nuevo Menú

AL presionar el botón grabar nos aparece un mensaje de que la grabación a la base fue exitosa, luego presionamos aceptar

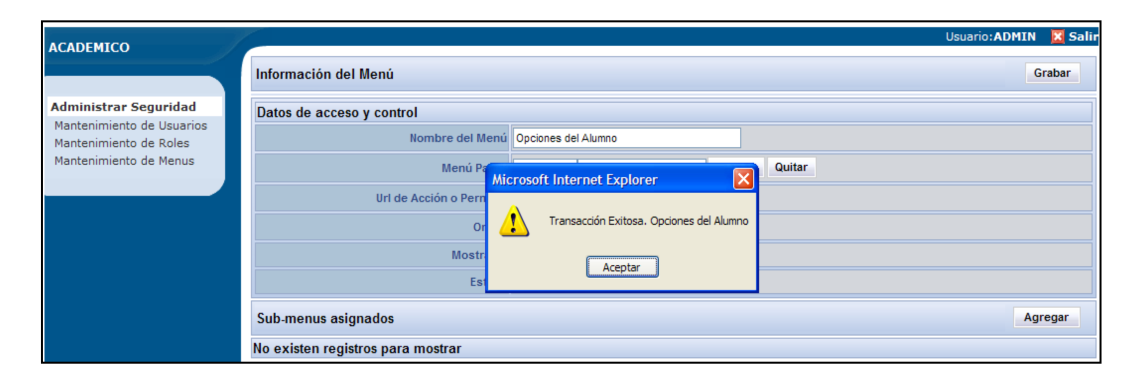


Fig. 2.23.3 Creación de un nuevo menú

2.3.1.8 Creación de un Submenú

Es similar a la creación de un menú principal solamente que se llena el campo de menú padre y el nivel será 1, 2, …..n, según el numero de submenús.



Fig. 2.24.1 Creacion de Submenu

Se ingresa el nombre del submenú y se busca el padre del mismo, escogemos el menú padre dando clic en el código.

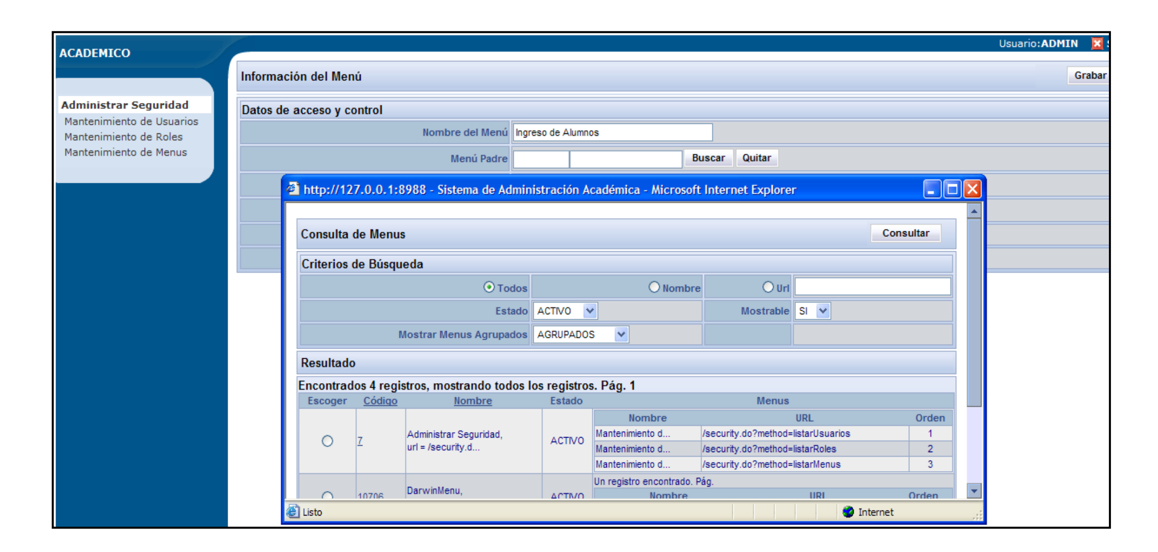


Fig. 2.24.2 Creacion de submenu

Se debe colocar el url de acción o permiso como en el ejemplo /web/jsp/alumno/Ingreso de alumno/* donde:

/web/jsp es la ruta de la carpeta donde se guardaran los jsp

/alumno es el padre del submenú

/Ingreso de Alumno/* es el submenú a crear del menú principal.

Luego de llenar todos los parámetros requeridos presionamos grabar y desde luego mostrara mensaje de transacción exitosa.

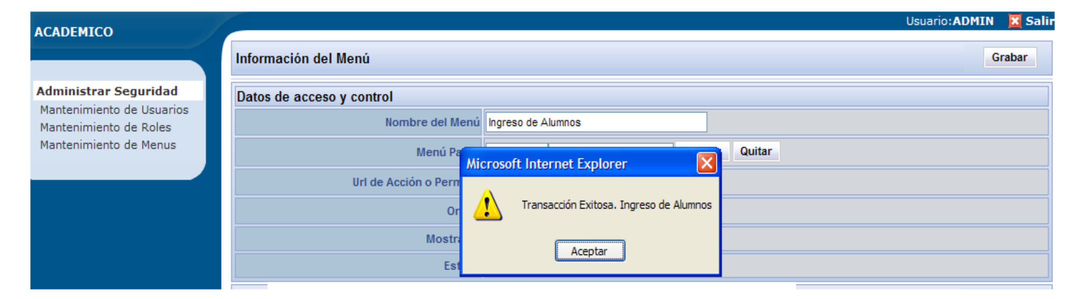


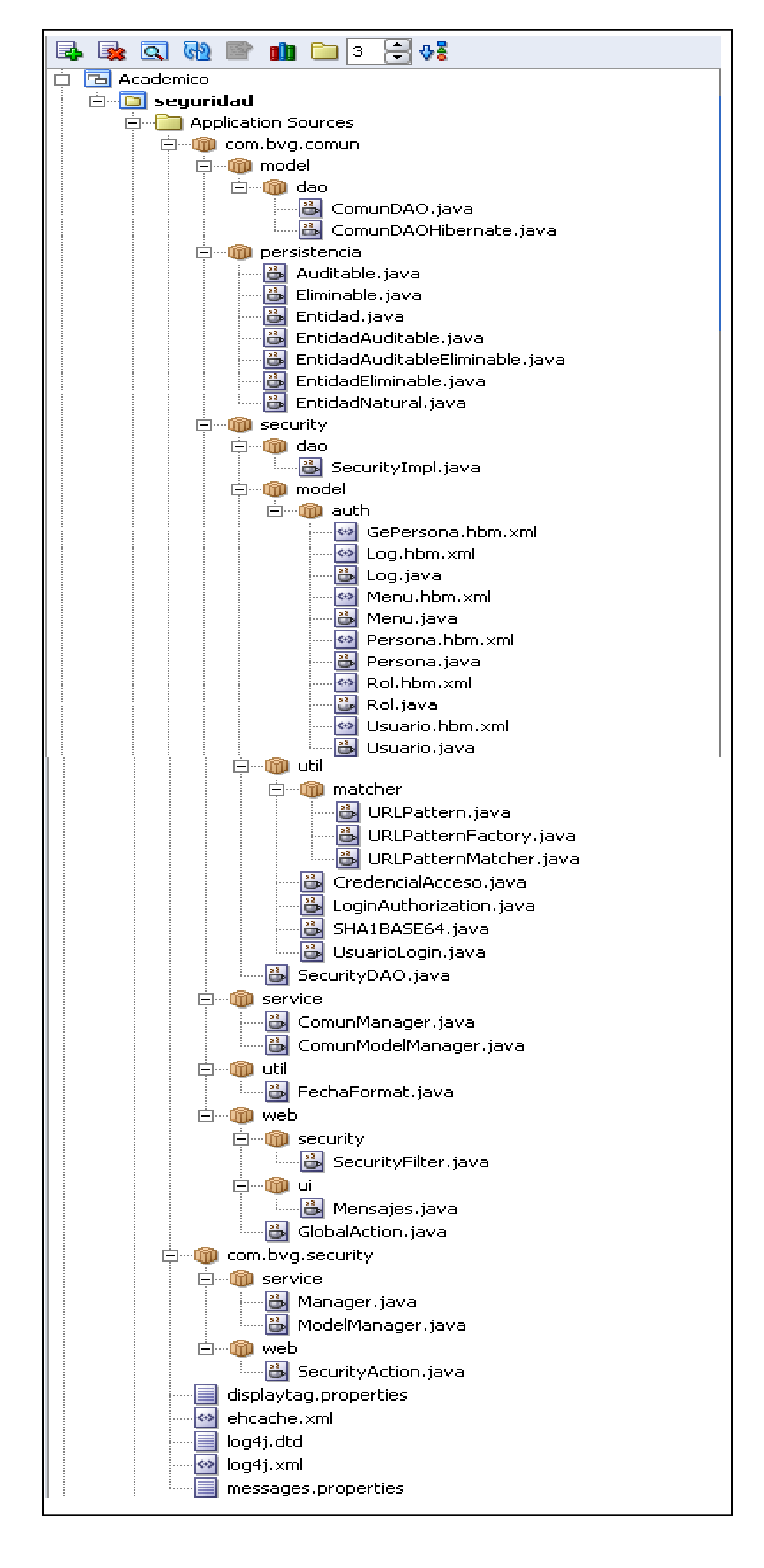
Fig. 2.24.3 Creacion de Submenu

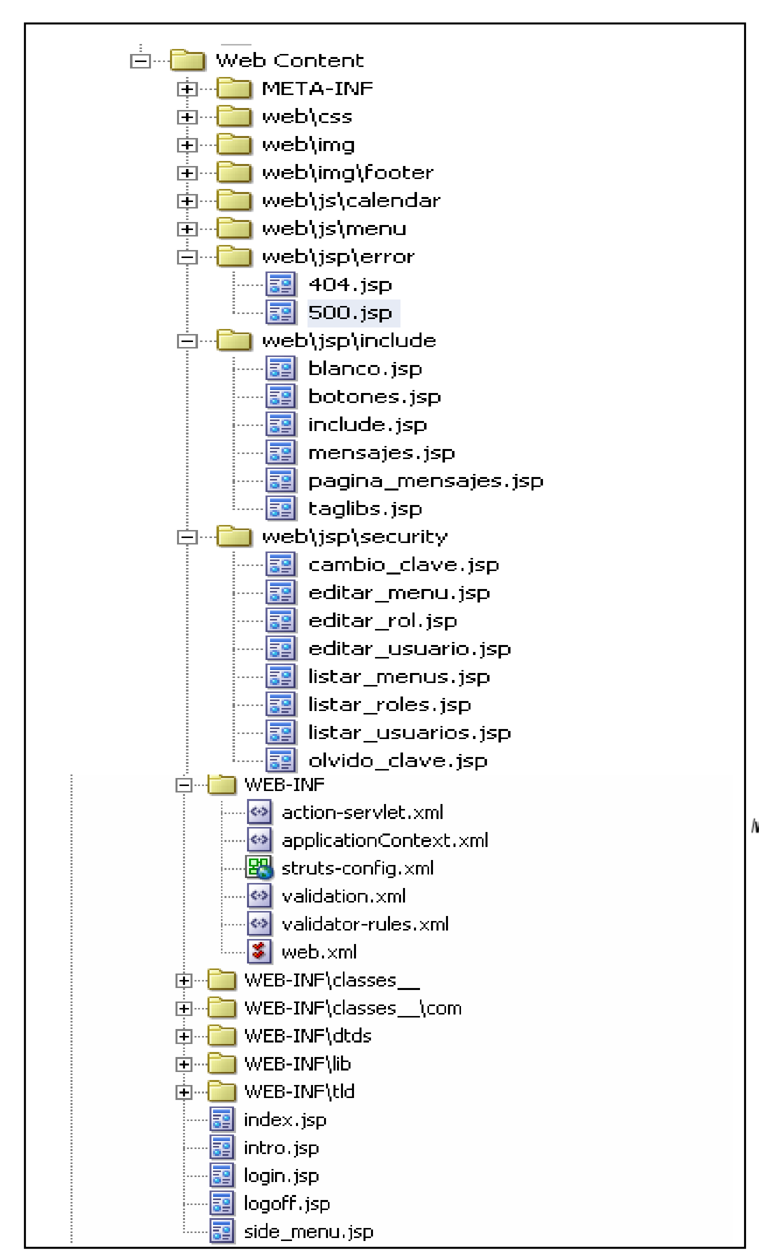
Visualización Creación de un Menú y sus respectivos Submenús

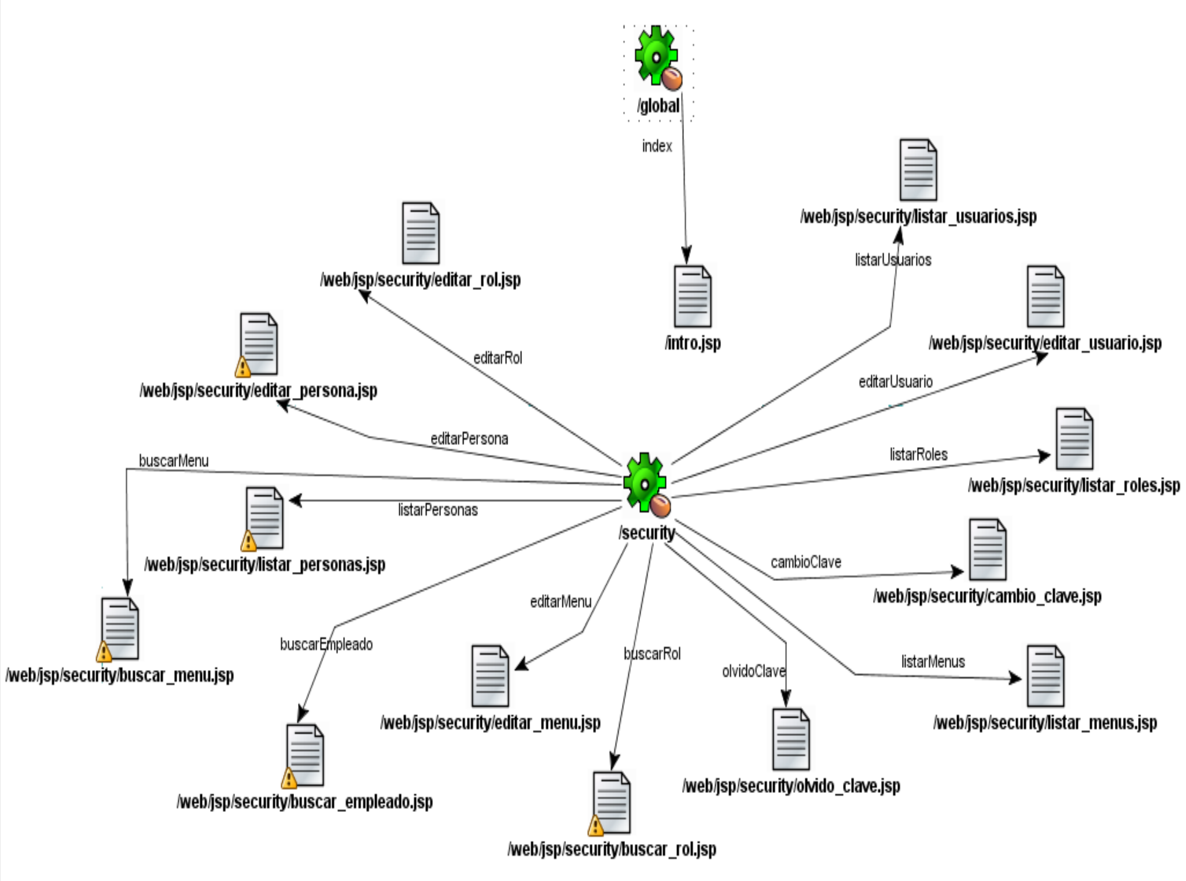
Fig. 2.24.4 Visualización Creación de un Menú y sus respectivos Submenús

MANUAL TECNICO

Estructura del Proyecto




Web Content
META-INF
web\css
web\img
web\img\footer
web\js\calendar
web\js\menu
web\jsp\error
404.jsp
500.jsp
web\jsp\include
blanco.jsp
botones.jsp
include.jsp
mensajes.jsp
pagina_mensajes.jsp
taglibs.jsp
web\jsp\security
cambio_clave.jsp
editar_menu.jsp
editar_rol.jsp
editar_usuario.jsp
listar_menus.jsp
listar_roles.jsp
listar_usuarios.jsp
olvido_clave.jsp
WEB-INF
action-servlet.xml
applicationContext.xml
struts-config.xml
validation.xml
validator-rules.xml
web.xml
WEB-INF\classes__
WEB-INF\classes__\com
WEB-INF\dtds
WEB-INF\lib
WEB-INF\tld
index.jsp
intro.jsp
login.jsp
logoff.jsp
side_menu.jsp


/global
index
/intro.jsp
/security
/web/jsp/security/listar_usuarios.jsp
listarUsuarios
/web/jsp/security/editar_rol.jsp
editarRol
/web/jsp/security/editar_usuario.jsp
editarUsuario
/web/jsp/security/editar_persona.jsp
editarPersona
buscarMenu
listarRoles
/web/jsp/security/listar_roles.jsp
listarPersonas
/web/jsp/security/listar_personas.jsp
cambioClave
/web/jsp/security/cambio_clave.jsp
editarMenu
/web/jsp/security/buscar_menu.jsp
buscarEmpleado
buscarRol
olvidoClave
listarMenus
/web/jsp/security/editar_menu.jsp
/web/jsp/security/listar_menus.jsp
/web/jsp/security/olvido_clave.jsp
/web/jsp/security/buscar_empleado.jsp
/web/jsp/security/buscar_rol.jsp

ComunDao.java

```
package com.bvg.comun.model.dao;
import java.util.HashMap;
import java.util.List;
import org.springframework.dao.DataAccessException;
import com.bvg.comun.security.model.auth.Menu;
import com.bvg.comun.security.model.auth.Rol;
import com.bvg.comun.security.model.auth.Usuario;

public interface ComunDAO {
        public void guardarObjeto(Object objeto) throws DataAccessException;
        public void borrarObjeto(Object objeto) throws DataAccessException;
```

Llamada a Procedimientos Generales para enlistar los menues, usuarios y roles.

```
        public List getMenus(HashMap filtros) throws DataAccessException;
        public List getUsuarios(HashMap filtros);
        public List getRoles(HashMap filtros);
```

Llamada a Procedimientos Particulares para buscar un usuario, un usuario asociado a un rol, un rol asociado a un menú y un menú.

```
        public Usuario findUsuarioById(String string) throws DataAccessException;
        public Usuario findUsuarioByIdWhithRoles(String usuarioLogin) throws
           DataAccessException;
        public Rol findRolWithMenus(Long id)throws DataAccessException;
        public Menu findMenuById(Long id);
}
```

ComunDAOHibernate.java

```
package com.bvg.comun.model.dao;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import org.springframework.dao.DataAccessException;
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
import com.bvg.comun.persistencia.Eliminable;
import com.bvg.comun.security.model.auth.Menu;
import com.bvg.comun.security.model.auth.Rol;
import com.bvg.comun.security.model.auth.Usuario;

public class ComunDAOHibernate extends HibernateDaoSupport implements ComunDAO {
public void guardarObjeto(Object objeto) throws DataAccessException{
  getHibernateTemplate().saveOrUpdate(objeto);
}

public void borrarObjeto(Object objeto) throws DataAccessException{
   getHibernateTemplate().delete(objeto);
}
```

Procedimientos Generales, para listar los menues, usurios y roles con sus respectivas validaciones.

```
public List getMenus(HashMap filtros) throws DataAccessException{
  List result = null;
  StringBuffer query = new StringBuffer("from Menu o ");
  query.append("left outer join fetch o.menus as submenu ");
  result = getHibernateTemplate().find(query.toString()+" order by o.nombre");
  return result;
}

public List getUsuarios(HashMap filtros){
  List result = null;
  StringBuffer query = new StringBuffer("from Usuario o ");
  result = getHibernateTemplate().find(query.toString()+" order by   o.id");
  return result;
}

public List getRoles(HashMap filtros){
  List result = null;
  StringBuffer query = new StringBuffer("from Rol o ");
  result = getHibernateTemplate().find(query.toString()+" order by o.id");
  return result;
}
```

Llamada a Procedimientos Particulares para buscar un usuario, un usuario asociado a un rol, un rol asociado a un menú y un menú.

```
public Usuario findUsuarioById(String id) throws DataAccessException{
  StringBuffer query = new StringBuffer("from Usuario o ");
  query.append("left outer join fetch o.roles ");
  List result = getHibernateTemplate().find(query.toString());
  if (result != null && result.size() > 0){
    for (Iterator iter = result.iterator(); iter.hasNext();){
      Usuario usuario = (Usuario) iter.next();
      if(usuario.getId().compareTo(id)== 0 ){
        return usuario;
      }
    }
  }
  return null;
}
public Usuario findUsuarioByIdWhithRoles(String usuarioLogin) throws
  DataAccessException{
  List lista = getHibernateTemplate().find("from Usuario user left outer  join
  fetch user.roles");
  if (lista != null && lista.size() > 0){
    for (Iterator iter = lista.iterator(); iter.hasNext();){
      Usuario usuario = (Usuario) iter.next();
      if(usuario.getId().compareTo(usuarioLogin)== 0 &&  usuario.getEstado()
        .compareTo(Eliminable.ESTADO_ACTIVO)==0){
       for (Iterator iterator =  usuario.getRoles().iterator(); iterator.hasNext();){
          Rol rol = (Rol) iterator.next();
          System.out.println(rol.getDescripcion());
         if  (rol.getEstado().compareTo(Eliminable.ESTADO_ACTIVO)==0){
```

```
                  for (Iterator iterator3 = rol.getMenus().iterator(); iterator3
                   .hasNext();) {
                     Menu menu = (Menu) iterator3.next();
                       System.out.println(menu.getUrl());
              }
           }
      }
            System.out.println(usuario.getPersona());
            return usuario;
    }
  }
 }
return null;
}

public Rol findRolWithMenus(Long id) throws DataAccessException {
   StringBuffer query = new StringBuffer("from Rol o ");
   query.append("left outer join fetch o.menus ");

   List result = getHibernateTemplate().find(query.toString());
   if (result != null && result.size() > 0){
      for (Iterator iter = result.iterator(); iter.hasNext();){
         Rol rol = (Rol) iter.next();
            if(rol.getId().compareTo(id)== 0 ){
               return rol;
          }
       }
       return (Rol) result.get(0);
   }
   return null;
}

public Menu findMenuById(Long id){
   StringBuffer query = new StringBuffer("from Menu o ");
   query.append("left outer join fetch o.menus ");
   List result = getHibernateTemplate().find(query.toString());
   if (result != null && result.size() > 0){
      for (Iterator iter = result.iterator(); iter.hasNext();){
         Menu menu = (Menu) iter.next();
         if(menu.getId().compareTo(id)== 0 ){
            return menu;
         }
      }
   }
   return null;
  }
}
```

Auditable.java

```
package com.bvg.comun.persistencia;
import java.util.Date;
import com.bvg.comun.security.model.auth.Usuario;
```

```
public interface Auditable{
	public Date getFechaIngreso();
	public void setFechaIngreso(Date fecha);

	public Date getFechaModificacion();
	public void setFechaModificacion(Date fecha);

	public Usuario getUsuarioIngreso();
	public void setUsuarioIngreso(Usuario user);

	public Usuario getUsuarioModificacion();
	public void setUsuarioModificacion(Usuario user);
}
```

Eliminable.java

```
package com.bvg.comun.persistencia;

public interface Eliminable {
	public static final String ESTADO_ACTIVO = "A";
	public static final String ESTADO_INACTIVO = "I";
```

Propiedad de la Tecnología Hibernate, permite eliminar o no un objeto de acuerdo al estado

```
	public String getEstado();
	public void setEstado(String activo);
}
```

Entidad.java

Clase que hereda todos los String

```
package com.bvg.comun.persistencia;

public abstract class Entidad {
	protected Long id;
	public Entidad() {
	}
```

El parámetro id de una entidad debe ser implementado por cada clase

```
	public abstract Long getId();
	public void setId(Long id) {
		this.id = id;
	}

	public boolean isNew() {
		return (this.id == null);
	}

	public String toString() {
		if (id != null)	return id.toString();
		else return this.getClass().getName();
	}
}
```

EntidadAuditable.java

Propiedades de Hibernate que permite Auditar o no un usuario como la fecha de ingreso, fecha de modificacion, etc.

```
package com.bvg.comun.persistencia;
import java.util.Date;
import com.bvg.comun.security.model.auth.Usuario;

public abstract class EntidadAuditable extends Entidad implements Auditable{
    protected Date fechaModificacion;
    protected Date fechaIngreso;
    protected Usuario usuarioIngreso;
    protected Usuario usuarioModificacion;

  public Date getFechaIngreso(){
        return this.fechaIngreso;
  }
  public void setFechaIngreso(Date fecha){
        this.fechaIngreso = fecha;
  }

  public Date getFechaModificacion(){
        return this.fechaModificacion;
  }

  public void setFechaModificacion(Date fecha) {
        this.fechaModificacion = fecha;
  }
  public Usuario getUsuarioIngreso() {
        return this.usuarioIngreso;
 }

    public void setUsuarioIngreso(Usuario user) {
        this.usuarioIngreso = user;
    }

   public Usuario getUsuarioModificacion() {
        return this.usuarioModificacion;
     }

    public void setUsuarioModificacion(Usuario user) {
        this.usuarioModificacion = user;
    }
}
```

EntidadAuditableEliminable.java

```
package com.bvg.comun.persistencia;

public abstract class EntidadAuditableEliminable extends EntidadAuditable implements
Eliminable {
            protected String estado;
```

```
	public String getEstado() {
		return estado;
	}

	public void setEstado(String activo) {
		estado = activo;
	}
}
```

EntidadEliminable.java

```
package com.bvg.comun.persistencia;

public abstract class EntidadEliminable extends Entidad implements Eliminable {
	protected String estado;

	public String getEstado() {
		return estado;
	}
	public void setEstado(String activo) {
		estado = activo;
	}
}
```

EntidadNatural.java

```
package com.bvg.comun.persistencia;

public abstract class EntidadNatural {
	protected String id;

	public EntidadNatural() {
	}

	public abstract String getId();

	public void setId(String id) {
		this.id = id;
	}

	public boolean isNew() {
		return (this.id == null);
	}
}
```

SecurityImpl.java

```
package com.bvg.comun.security.dao;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import org.springframework.dao.DataAccessException;
```

```
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
import com.bvg.comun.persistencia.Eliminable;
import com.bvg.comun.security.SecurityDAO;
import com.bvg.comun.security.model.auth.Log;
import com.bvg.comun.security.model.auth.Menu;
import com.bvg.comun.security.model.auth.Rol;
import com.bvg.comun.security.model.auth.Usuario;
import com.bvg.comun.security.util.UsuarioLogin;

public class SecurityImpl extends HibernateDaoSupport implements SecurityDAO {
  public Usuario findUsuario(String usrLogin) throws DataAccessException {
     List lista;
     lista = getHibernateTemplate().find("from Usuario usr ");
     if (lista.size() > 0 ) {
        for (Iterator iter = lista.iterator(); iter.hasNext();) {
          Usuario usuario = (Usuario) iter.next();
            if(usuario.getId().compareTo(usrLogin)== 0 &&
           usuario.getEstado().compareTo(Eliminable.ESTADO_ACTIVO)==0)
            {
               return usuario;
            }
          }
     }
     return null;
     }

    public UsuarioLogin getUsuario(String userlogin) {
          return findUsuario(userlogin);
   }

   public Menu getOpcion(String id_opcion) {
          return (Menu) getHibernateTemplate().get(Menu.class,
               Integer.valueOf(id_opcion));
  }

  public List getOpciones(String userlogin) {
          List lista = getHibernateTemplate().find("from Usuario user left outer
            join fetch user.roles");
          List result = new ArrayList();
          if (lista != null && lista.size() > 0) {
             List roles = getHibernateTemplate().find("from Rol rol left outer join
                fetch rol.menus");
             if (roles != null && roles.size() > 0) {
                   for (Iterator iter = lista.iterator(); iter.hasNext();) {
                      Usuario usuario = (Usuario) iter.next();
           if(usuario.getId().compareTo(userlogin)== 0 &&
                  usuario.getEstado().compareTo(Eliminable.ESTADO_ACTIVO)==0) {
                       for (Iterator iterator = usuario.getRoles().iterator(); iterator
                          .hasNext();) {
                          Rol rol = (Rol) iterator.next();
                          for (Iterator iterator2 = roles.iterator(); iterator2.hasNext();){
                             Rol rol2 = (Rol) iterator2.next();
                          if (rol.getId().compareTo(rol2.getId())== 0 &&
```

```
                        rol.getEstado().compareTo(Eliminable.ESTADO_ACTIVO)
                            ==0)  {
                         for (Iterator iterator4 = rol2.getMenus().iterator(); iterator4
                                .hasNext();) {
                                Menu menu = (Menu) iterator4.next();
                    if(menu.getEstado().compareTo(Eliminable.ESTADO_ACTIVO)
                     == 0){
                        result.add(menu);
                     }
                  }
                }
              }
            }
            break;
         }
        }
      }
    }
    return result;
}
 public void saveLog(Log log) {
     getHibernateTemplate().save(log);
 }
}
```

Log.hbm.xml

Mapeo de Tabla LOG y UID

```
<hibernate-mapping
>
    <class
        name="com.bvg.comun.security.model.auth.Log"
        table="LOG"
    >
    <id
            name="id"
            column="codigolog"
            type="java.lang.Long"
     >
            <generator class="hilo">
                <param name="table">UID</param>
                <param name="column">codigolog</param>
                <param name="max_lo">100</param>
            </generator>
        </id>
    <property
            name="fecha"
            type="java.util.Date"
            update="true"
            insert="true"
            column="fecharequerimiento"
            not-null="true"
        />
```

```
      <property
          name="host"
          type="java.lang.String"
          update="true"
          insert="true"
          column="hostremoto"
      />
      <property
          name="url"
          type="java.lang.String"
          update="true"
          insert="true"
          column="urlrequerido"
          not-null="true"
      />
      <many-to-one
          name="usuario"
          class="com.bvg.comun.security.model.auth.Usuario"
          cascade="none"
          outer-join="auto"
          update="true"
          insert="true"
          column="codigousuario"
          not-null="true"
      />
      <property
          name="addr"
          type="java.lang.String"
          update="true"
          insert="true"
          column="addrremoto"
      />
      <property
          name="user"
          type="java.lang.String"
          update="true"
          insert="true"
          column="userremoto"
      />
   </class>
</hibernate-mapping>
```

Log.java

Clase que me ayuda a obtener los movimientos de un usuario tales como la fecha en que ingreso, a donde acceso y desde donde acceso. Estos movimientos se guarda en la base de datos para su respectiva revision.

```
package com.bvg.comun.security.model.auth;
import java.util.Date;
import com.bvg.comun.persistencia.Entidad;

public class Log extends Entidad {
        private Usuario usuario;
```

```
private String url;
private Date fecha;
private String host;
private String user;
private String addr;

public Log() {
	super();
}

public Long getId() {
	return id;
}

public Date getFecha() {
	return fecha;
}

public void setFecha(Date fecha) {
	this.fecha = fecha;
}

public String getHost() {
	return host;
}

public void setHost(String host) {
	this.host = host;
}

public String getUrl() {
	return url;
}

public void setUrl(String url) {
	this.url = url;
}

public Usuario getUsuario() {
	return usuario;
}

public void setUsuario(Usuario usuario) {
	this.usuario = usuario;
}

public String getAddr() {
	return addr;
}

public void setAddr(String addr) {
	this.addr = addr;
}
```

```
		public String getUser() {
			return user;
		}

		public void setUser(String user) {
			this.user = user;
		}
}
```

Menu.hbm.xml

Mapeo tabla MENU

```
<hibernate-mapping
>
  <class
    name="com.bvg.comun.security.model.auth.Menu"
    table="MENU"
  >
  <id
      name="id"
      column="codigomenu"
      type="java.lang.Long"
    >
      <generator class="hilo">
        <param name="table">UID</param>
        <param name="column">codigomenu</param>
        <param name="max_lo">100</param>
      </generator>
    </id>
    <property
      name="url"
      type="java.lang.String"
      update="true"
      insert="true"
      column="url"
      not-null="true"
    />
    <property
      name="estado"
      type="java.lang.String"
      update="true"
      insert="true"
      column="estado"
      not-null="true"
    />
    <property
      name="orden"
      type="java.lang.Integer"
      update="true"
      insert="true"
      column="orden"
      not-null="true"
    />
```

```
        <property
           name="idPadre"
           type="java.lang.Long"
           update="true"
           insert="true"
           column="codigomenupadre"
        />
        <property
           name="nombre"
           type="java.lang.String"
           update="true"
           insert="true"
           column="nombre"
           not-null="true"
        />
        <set
           name="menus"
           lazy="false"
           inverse="true"
           cascade="save-update"
           sort="unsorted"
           order-by="orden"
        >
        <key
              column="codigomenupadre"
           >
           </key>
           <one-to-many
                 class="com.bvg.comun.security.model.auth.Menu"
           />
        </set>
        <property
           name="mostrable"
           type="java.lang.String"
           update="true"
           insert="true"
           column="mostrable"
           not-null="true"
        />
    </class>
</hibernate-mapping>
```

Menu.java

```
package com.bvg.comun.security.model.auth;
import java.util.Set;
import com.bvg.comun.persistencia.EntidadEliminable;
import com.bvg.comun.security.util.CredencialAcceso;

public class Menu extends EntidadEliminable implements CredencialAcceso, Comparable{
        public static final String MOSTRABLE = "S";
        public static final String NO_MOSTRABLE = "N";
```

Fields

```
private String nombre;
private String url;
private Integer orden;
private Long idPadre;
private Set menus;
private String mostrable;
```

Constructor por default

```
public Menu()  {
}

public Long getId() {
  return id;
}

 public String getPattern() {
   return url;
}

public void setUrl(String url) {
   this.url = url;
}

public String getUrl() {
  return url;
}

public void setEstado(String estado) {
   this.estado = estado;
}

public String getEstado() {
  return estado;
}
public void setOrden(Integer orden) {
   this.orden = orden;
}

public Integer getOrden() {
  return orden;
}

public int compareTo(Object o) {
  if (o != null &&o instanceof Menu) {
    Menu anotherMenu = (Menu) o;
    return this.getOrden().compareTo(anotherMenu.getOrden());
  }
  return 0;
}

public void setNombre(String nombre) {
  this.nombre = nombre; }
```

```
public Long getIdPadre() {
  return idPadre;
}

public void setIdPadre(Long idPadre) {
  this.idPadre = idPadre;
}

public String getNombre() {
  return nombre;
}

public Set getMenus() {
  return menus;
}

public void setMenus(Set menus) {
  this.menus = menus;
}

public String getMostrable() {
  return mostrable;
}

public void setMostrable(String mostrable) {
  this.mostrable = mostrable;
}

public boolean isMenuMostrable(){
  if (mostrable != null) {
    if(mostrable.compareTo( Menu.MOSTRABLE )== 0) {
      return true;
    }else
        if(mostrable.compareTo( Menu.NO_MOSTRABLE )== 0){
          return false;
        }
    }
    return false;
  }
}
```

Rol.hbm.xml

Mapeo de la tabla ROL

```
<hibernate-mapping
>
    <class
        name="com.bvg.comun.security.model.auth.Rol"
        table="ROL"
    >
     <id
          name="id"
```

```
        column="codigorol"
        type="java.lang.Long"
    >
        <generator class="hilo">
            <param name="table">uid</param>
            <param name="column">codigorol</param>
            <param name="max_lo">100</param>
        </generator>
    </id>
    <set
        name="menus"
        table="ROLMENU"
        lazy="false"
        cascade="save-update"
        sort="unsorted"
        order-by="codigomenu asc"
    >
    <key
            column="codigorol"
     >
        </key>
       <many-to-many
            class="com.bvg.comun.security.model.auth.Menu"
            column="codigomenu"
            outer-join="auto"
         />
    </set>
    <property
        name="nombre"
        type="java.lang.String"
        update="true"
        insert="true"
        column="nombre"
        not-null="true"
    />
    <property
        name="descripcion"
        type="java.lang.String"
        update="true"
        insert="true"
        column="descripcion"
    />
    <property
        name="estado"
        type="java.lang.String"
        update="true"
        insert="true"
        column="estado"
        not-null="true"
    />
  </class>
</hibernate-mapping>
```

Rol.java

```
package com.bvg.comun.security.model.auth;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;
import com.bvg.comun.persistencia.EntidadEliminable;

public class Rol extends EntidadEliminable {
  private String nombre;
  private String descripcion;
  private Set menus = new HashSet();

  public Rol() {
    super();
  }

  public Long getId()  {
    return id;
  }

public Set getMenus() {
  return menus;
}

public void setMenus(Set menus) {
  this.menus = menus;
}

public void addMenu(Menu o){
  for (Iterator iter = menus.iterator(); iter.hasNext();) {
    Menu op = (Menu) iter.next();
    if (op.getId().compareTo(o.getId()) == 0) {
      return;  }
  }   menus.add(o);
  }

public String getNombre() {
  return nombre;
}

public void setNombre(String nombre) {
  this.nombre = nombre;
}

public String getDescripcion() {
  return descripcion;
}

public void setDescripcion(String descripcion) {
  this.descripcion = descripcion;
}
}
```

Usuario.hbm.xml

Mapeo de la tabla USUARIOS

```
<hibernate-mapping
>
  <class
    name="com.bvg.comun.security.model.auth.Usuario"
    table="USUARIOS"
  >
    <id
      name="id"
      column="codigousuario"
      type="java.lang.String"
    >
      <generator class="assigned">
       </generator>
    </id>
    <property
      name="clave"
      type="java.lang.String"
      update="true"
      insert="true"
      column="clave"
      not-null="true"
    />
    <property
      name="estado"
      type="java.lang.String"
      update="true"
      insert="true"
      column="estado"
      not-null="true"
    />
    <set
      name="roles"
      table="USUARIOROL"
      lazy="false"
      cascade="save-update"
      sort="unsorted"
      order-by="codigorol asc"
    >
    <key
        column="codigousuario"
      >
      </key>
      <many-to-many
        class="com.bvg.comun.security.model.auth.Rol"
        column="codigorol"
        outer-join="auto"
       />
    </set>
    <property
      name="nombre"
```

```
            type="java.lang.String"
            update="true"
            insert="true"
            column="nombre"
        />
        <property
            name="pregunta"
            type="java.lang.String"
            update="true"
            insert="true"
            column="pregunta"
        />
       <property
            name="respuesta"
            type="java.lang.String"
            update="true"
            insert="true"
            column="respuesta"
        />
        <property
            name="nombreUsuario"
            type="java.lang.String"
            update="true"
            insert="true"
            column="usuario"
        />
        <property
            name="fechaSolicitud"
            type="java.util.Date"
            update="true"
            insert="true"
            column="fechasolicitud"
        />
        <property
            name="estadoCambio"
            type="java.lang.String"
            update="true"
            insert="true"
            column="estadocambio"
        />
        <property
            name="fechaCambioClave"
            type="java.util.Date"
            update="true"
            insert="true"
            column="fechacambioclave"
        />  </class>
</hibernate-mapping>
```

Usuario.java

```
package com.bvg.comun.security.model.auth;
import java.util.ArrayList;
```

```
import java.util.Collection;
import java.util.Date;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Set;
import com.bvg.comun.persistencia.Eliminable;
import com.bvg.comun.persistencia.EntidadNatural;
import com.bvg.comun.security.util.SHA1BASE64;
import com.bvg.comun.security.util.UsuarioLogin;

public class Usuario extends EntidadNatural implements UsuarioLogin, Eliminable{
public static final String INICIAL = "I";
public static final String CLAVE_CAMBIADA = "C";
```

Fields

```
private String clave;
private String estado;
private String nombre;
private String pregunta;
private String respuesta;
private Date fechaSolicitud;
private Set roles = new HashSet();
private Persona persona;
private transient String confirmClave;
private String estadoCambio;
private Date fechaCambioClave;
```

Constructor por default

```
public Usuario() {
}

public Usuario(String id) {
  this.id = id;
}

public String getId() {
  return this.id;
}

public void setId(String id) {
  this.id = id;
}

public void setClave(String clave) {
  this.clave = clave;
}

public String getClave() {
  return clave;
}

public void setEstado(String estado) {
```

```
    this.estado = estado;
  }

  public String getEstado() {
    return estado;
  }

  public void setRoles(Set roles) {
    this.roles = roles;
  }

  public Set getRoles() {
    return roles;
  }

  public Collection getCredenciales() {
    List opciones = new ArrayList();
    for (Iterator iter = roles.iterator(); iter.hasNext();) {
      Rol rol = (Rol) iter.next();
      for (Iterator iterator = rol.getMenus().iterator(); iterator.hasNext();) {
        Menu opcion = (Menu) iterator.next();
        opciones.add(opcion);
      }
    }
    return opciones;
  }

  public String getLoginName() {
    return getId();
  }

  public boolean validaPassword(String intento) {
    String usrPasswd = getClave();
    String intentoSha = SHA1BASE64.encriptar(intento);
    if (usrPasswd != null) {
      return usrPasswd.trim().equals(intentoSha);
    }
      return false;
    }

    public boolean isActivo() {
      return (getEstado()!= null &&
      getEstado().compareTo(Eliminable.ESTADO_ACTIVO)== 0)? true : false;
    }

   public Persona getPersona() {
      return persona;
    }

    public void setPersona(Persona persona) {
      this.persona = persona;
    }
```

```
public String getNombre() {
  return nombre;
}

public void setNombre(String nombre) {
  this.nombre = nombre;
}

public String getPregunta() {
  return pregunta;
}

public void setPregunta(String pregunta) {
  this.pregunta = pregunta;
}

public String getRespuesta() {
  return respuesta;
}

public void setRespuesta(String respuesta) {
  this.respuesta = respuesta;
}

public Date getFechaSolicitud() {
  return fechaSolicitud;
}

public void setFechaSolicitud(Date fechaSolicitud) {
  this.fechaSolicitud = fechaSolicitud;
}

public String getConfirmClave() {
  return confirmClave;
}

public void setConfirmClave(String confirmClave) {
  this.confirmClave = confirmClave;
}

public String getEstadoCambio() {
  return estadoCambio;
}

public void setEstadoCambio(String estadoCambio) {
  this.estadoCambio = estadoCambio;
}

public Date getFechaCambioClave() {
  return fechaCambioClave;
}

public void setFechaCambioClave(Date fechaCambioClave) {
```

```
    this.fechaCambioClave = fechaCambioClave;
  }

  public void setNombreUsuario(String nombreUsuario) {

   }

  public String getNombreUsuario() {
      return this.getId();
  }
}
```

URLPattern.java

URLPattern contiene las comparaciones de los URL pattern, las asociaciones de los SecurityConstraint y la colección de objetos para esos patterns.

```
package com.bvg.comun.security.util.matcher;
import org.apache.oro.text.regex.Pattern;
import org.apache.oro.text.regex.PatternCompiler;
import org.apache.oro.text.regex.Perl5Compiler;

public class URLPattern implements Comparable {
  public static final int EXACT_TYPE = 1;
  public static final int PATH_TYPE = 2;
  public static final int EXTENSION_TYPE = 3;
  public static final int DEFAULT_TYPE = 4;
  protected String pattern;
  protected String convertedPattern;
  protected Pattern compiledPattern;
  protected int order;
  protected int patternType;
  protected int pathLength;

public URLPattern(
    String pattern,
    int order,
    PatternCompiler compiler ) throws Exception {
    this.pattern = pattern;
    this.order = order;
    initPatternType();
    initPathLength();
    initConvertedPattern();
    initCompiledPattern(compiler);
}

public String getPattern() {
  return pattern;
}

public Pattern getCompiledPattern() {
  return compiledPattern;
}
```

```
public int getPatternType() {
  return patternType;
}

public int getPathLength() {
  return pathLength;
}

public int getOrder() {
  return order;
}

protected void initPatternType() {
  if ("/".equals(pattern)) {
    patternType = DEFAULT_TYPE;
  } else
    if (pattern.startsWith("*.")) {
      patternType = EXTENSION_TYPE;
    } else
      if (pattern.startsWith("/") && pattern.endsWith("/*"))  {
        patternType = PATH_TYPE;
      } else {
        patternType = EXACT_TYPE;
    }
  }

  protected void initPathLength() {
    pathLength = -1;
    int pos = pattern.indexOf('/');
    while (pos != -1) {
      pathLength++;
      pos = pattern.indexOf('/', pos + 1);
    }
  }
  protected void initConvertedPattern()  {
    if (patternType == DEFAULT_TYPE) {
      convertedPattern = ".*";
    } else {
      StringBuffer buf = new StringBuffer(pattern);
      int pos;
      pos = buf.toString().indexOf('.');
      while (pos != -1) {
        buf.insert(pos, "\\");
        pos = buf.toString().indexOf('.', pos + 2);
      }
      pos = buf.toString().indexOf('*');
      while (pos != -1) {
        buf.replace(pos, pos + 1, ".*");
        pos = buf.toString().indexOf('*', pos + 2);
      }
      pos = buf.toString().indexOf('/');
      while (pos != -1) {
        buf.replace(pos, pos + 1, "/+");
```

```
          pos = buf.toString().indexOf('/', pos + 2);
        }
        switch (patternType) {
          case PATH_TYPE:
          buf.insert(0, '^');
          buf.insert(buf.length()-4, "(");
          buf.append(")?$");
          break;
          case EXTENSION_TYPE:
            buf.append('$');
            break;
          case EXACT_TYPE:
            buf.insert(0, '^');
            buf.append('$');
            break;
        }
        convertedPattern = buf.toString();
      }
    }

protected void initCompiledPattern(PatternCompiler compiler) throws Exception {
     compiledPattern = compiler.compile(convertedPattern,
     Perl5Compiler.READ_ONLY_MASK);
}

public int compareTo(Object obj) throws ClassCastException{
    URLPattern other = (URLPattern) obj;
    if (this.equals(other)) {
      return 0;
    }
    int c = patternType - other.patternType;
    if (c == 0) {
      switch (patternType) {
        case PATH_TYPE:
          c = other.pathLength - pathLength;
          if (c != 0) {
            break;
    }
        case EXACT_TYPE:
        case EXTENSION_TYPE:
        case DEFAULT_TYPE:
        c = order - other.order;
      }
    }
    return c;
  }
}
```

URLPatternFactory.java

URLPatternFactory crea instancias de URLPattern

```
package com.bvg.comun.security.util.matcher;
import org.apache.oro.text.regex.PatternCompiler;
```

```
import org.apache.oro.text.regex.Perl5Compiler;
public class URLPatternFactory {
  protected PatternCompiler compiler;

public URLPatternFactory() {
    compiler = new Perl5Compiler();
  }

  public URLPatternMatcher createURLPatternMatcher() {
    return new URLPatternMatcher();
  }
  public URLPattern createURLPattern(
    String pattern,
    int order ) throws Exception {
    return new URLPattern(pattern, order, compiler);
  }
}
```

URLPatternMatcher.java

URLPatternMatcher , guarda el objeto que va a aser usado para la combinacion requeridad de pattern con el URLPattern objects.

```
package com.bvg.comun.security.util.matcher;
import org.apache.oro.text.regex.PatternMatcher;
import org.apache.oro.text.regex.Perl5Matcher;

public class URLPatternMatcher {
  private PatternMatcher patternMatcher;

public URLPatternMatcher() {
  patternMatcher = new Perl5Matcher();
}

public boolean match(String pattern, URLPattern urlPattern)throws Exception{
  return patternMatcher.matches(pattern, urlPattern.getCompiledPattern());
}
}
```

CredencialAcceso.java

Devuelve el credencial de acceso a una opcion

```
package com.bvg.comun.security.util;

public interface CredencialAcceso{

public String getPattern();
}
```

LoginAuthorization.java

Devuelve un usuario si fue encontrado

```
package com.bvg.comun.security.util;
```

```
import java.util.List;
import com.bvg.comun.security.model.auth.Log;

public interface LoginAuthorization{
  public UsuarioLogin getUsuario(String userlogin);
  public List getOpciones(String userlogin);
  public void saveLog(Log log);
}
```

SHA1BASE64.java

Metodo que nos ayuda a encriptar las claves de cada usuario del sistema.

```
package com.bvg.comun.security.util;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import sun.misc.BASE64Encoder;

public class SHA1BASE64{
  public static String encriptar(String textoplano) throws IllegalStateException{
    MessageDigest md = null;
    try {
      md = MessageDigest.getInstance("SHA");
      Instancia de generador SHA-1
    } catch(NoSuchAlgorithmException e) {
      throw new IllegalStateException(e.getMessage());
    } try {
        md.update(textoplano.getBytes("UTF-8"));
        Generación de resumen de mensaje
    } catch(UnsupportedEncodingException e){
      throw new IllegalStateException(e.getMessage());
    }
    byte raw[] = md.digest(); // Obtención del resumen de mensaje
    String hash = (new BASE64Encoder()).encode(raw);
    Traducción a BASE64
    return hash;
   }
}
```

UsuarioLogin.java

Interface de login

```
package com.bvg.comun.security.util;
import java.util.Collection;

public interface UsuarioLogin{
        Devuelve el login de un usuario
        public String getLoginName();
        Valida el password de un usuario
        public boolean validaPassword(String intento);
        Indica si el usuario esta activo
        public boolean isActivo();
        Devuelve las credenciales
```

```
        public Collection getCredenciales();
}
```

SecurityDAO.java

```
package com.bvg.comun.security;
import org.springframework.dao.DataAccessException;
import com.bvg.comun.security.model.auth.Usuario;
import com.bvg.comun.security.util.LoginAuthorization;

public interface SecurityDAO extends LoginAuthorization {
  public Usuario findUsuario(String usrLogin) throws DataAccessException;
}
```

ComunManager.java

```
package com.bvg.comun.service;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import com.bvg.comun.model.dao.ComunDAO;
import com.bvg.comun.persistencia.Auditable;
import com.bvg.comun.persistencia.Eliminable;
import com.bvg.comun.persistencia.Entidad;
import com.bvg.comun.security.model.auth.Menu;
import com.bvg.comun.security.model.auth.Rol;
import com.bvg.comun.security.model.auth.Usuario;

public class ComunManager implements ComunModelManager {
  protected ComunDAO comunDAO;
  public ComunManager() {  }

  public void setComunDAO(ComunDAO comunDAO) {
    this.comunDAO = comunDAO;
  }

  public void guardarObjeto(Object objeto) {
    comunDAO.guardarObjeto(objeto);
  }

  public void guardarEntidad(Entidad entidad, String id_usuario) {
    if (entidad instanceof Auditable) {
      Auditable auditable = (Auditable) entidad;
      Usuario user = null;
      if (id_usuario != null) {
        user = comunDAO.findUsuarioById(id_usuario);
        }else{
          user = comunDAO.findUsuarioById("ADMIN");
        }
        if  (entidad.isNew() || auditable.getFechaIngreso() == null){
```

```
            auditable.setFechaIngreso(Calendar.getInstance().getTime());
            auditable.setUsuarioIngreso(user);
          }
          auditable.setFechaModificacion(Calendar.getInstance().getTime());
          auditable.setUsuarioModificacion(user);
        }
        if  (entidad.isNew() ) {
          if (entidad instanceof Eliminable) {
            Eliminable eliminable = (Eliminable) entidad;
            eliminable.setEstado(Eliminable.ESTADO_ACTIVO);                    }
        }
        comunDAO.guardarObjeto(entidad);
      }

      public Usuario findUsuarioByIdWhithRoles(String usuarioLogin) {
        return comunDAO.findUsuarioByIdWhithRoles(usuarioLogin);
      }

      public Usuario findUsuarioById(String loginName){
        return comunDAO.findUsuarioById(loginName);
      }

      public List getMenus(HashMap filtros) {
        List list = comunDAO.getMenus(filtros);
        Set set = new HashSet(list);
        list = new ArrayList(set);
        return list;
      }

      public List getUsuarios(HashMap filtros) {
           return comunDAO.getUsuarios(filtros);
      }

      public List getRoles(HashMap filtros) {
        return comunDAO.getRoles(filtros);
      }

      public Rol findRolWithMenus(String id) {
        return comunDAO.findRolWithMenus(Long.valueOf(id));
      }

      public Menu findMenuById(String id) {
        return comunDAO.findMenuById(Long.valueOf(id));
      }
}
```

SecurityFilter.java

Implementa la seguridad del sistema.

```
package com.bvg.comun.web.security;
import java.io.IOException;
import java.util.Calendar;
import java.util.Collection;
```

```
import java.util.Date;
import java.util.Iterator;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.context.ApplicationContext;
import org.springframework.web.context.support.WebApplicationContextUtils;
import com.bvg.comun.security.model.auth.Log;
import com.bvg.comun.security.model.auth.Usuario;
import com.bvg.comun.security.util.CredencialAcceso;
import com.bvg.comun.security.util.LoginAuthorization;
import com.bvg.comun.security.util.UsuarioLogin;
import com.bvg.comun.security.util.matcher.URLPattern;
import com.bvg.comun.security.util.matcher.URLPatternFactory;
import com.bvg.comun.security.util.matcher.URLPatternMatcher;

public class SecurityFilter implements Filter {
  Permite obtener acceso al LoginAuthorization
  FilterConfig filterConfig;
  Validacion de Opciones a través de patrones de RegEx
  URLPatternFactory urlPatternFactory;
  Nombre del Bean que tiene el acceso a datos
  String loginAuthorizationParam;
  Nombre del Usuario Invitado
  String loginGuestUser;
  final private static String SESSION_LOGGED = "w@L1_SESSION_LOGGED";
        private final static String SESSION_LAST_URI =
        "w@L1_SESSION_LAST_URI";

 Parametro de Inicializacion
  public final static String LOGIN_AUTHORIZATION_PARAM = "loginAuthorization";
   public final static String LOGIN_USUARIO_INVITADO_PARAM = "guestUser";
  Login del Usuario
  public final static String SESSION_USERLOGIN = "SESSION_USERLOGIN";
  Mensaje de Error al Logonearse y no autenticarse
  public final static String SECURITY_MSGLOGIN = "SECURITY_MSGLOGIN";
  private static final String LOG_OFF_URL = "/logoff.jsp";
  private static final String ACTION_URI = "/j_security_check";
  private static final String ROOT_URI = "/";
  private static final String WIDTHOUT_PERMISSIONS = "No tiene el suficiente
    nivel de acceso a este recurso";
  private static final String LOG_IN_URL = "/login.jsp";

  public void init(FilterConfig filterConfig) throws ServletException {
    this.filterConfig = filterConfig;
    loginAuthorizationParam =
      filterConfig.getInitParameter(LOGIN_AUTHORIZATION_PARAM);
        loginGuestUser =
```

```
      filterConfig.getInitParameter(LOGIN_USUARIO_INVITADO_PARAM);
      urlPatternFactory = new URLPatternFactory();
  }

  public void doFilter(ServletRequest req, ServletResponse res,
    FilterChain chain) throws IOException, ServletException {
    HttpServletRequest request = (HttpServletRequest) req;
    HttpServletResponse response = (HttpServletResponse) res;

   String uri = request.getRequestURI().substring(request.getContextPath().length());

    Collection opcionesInvitado = null;
    if (request.getSession().getAttribute("opInvitado") == null){
         opcionesInvitado = getLoginAuthorization().getOpciones(loginGuestUser);
         request.getSession().setAttribute("opInvitado", opcionesInvitado);
    }else{
         opcionesInvitado = (Collection)
         request.getSession().getAttribute("opInvitado");
    }

    for (Iterator iter = opcionesInvitado.iterator(); iter.hasNext();) {
         CredencialAcceso credencial = (CredencialAcceso) iter.next();
         if (matchURL(credencial.getPattern(), uri)) {
           System.out.println(credencial.getPattern());
           chain.doFilter(request, response);
            return;
     }
   }
```

Permitido el ingreso del usuario, debido a que estoy recuperando la clave

```
String isOlvido = request.getParameter("isOlvido");
  if (isOlvido != null && isOlvido.compareTo(request.getSession().getId())== 0){
    chain.doFilter(request, response);
    return;
  }
```

Permitido debido a que estoy cambiando la clave

```
String isCambio = (String) request.getSession()
  .getAttribute(Usuario.CLAVE_CAMBIADA);
   if (isCambio != null && isCambio.compareTo(Usuario.INICIAL) == 0){
     chain.doFilter(request, response);
         return;
   }
 Boolean islogged = (Boolean) request.getSession()
   .getAttribute(SESSION_LOGGED);
 if ((islogged != null) && islogged == Boolean.TRUE) {
   UsuarioLogin userLogin = (UsuarioLogin)request.getSession().
   getAttribute(SecurityFilter.SESSION_USERLOGIN);
 if (!requestRequiresLoggedSession(uri, userLogin.getLoginName()) ){
   int result = uri.indexOf("j_security_check");
   if (result != -1){
     request.getRequestDispatcher(request.getContextPath()).forward(request,
       response);
   }else {
     chain.doFilter(request, response); //Pass the request/response on
```

```
  }
    Log log = new Log();
    log.setFecha(Calendar.getInstance().getTime());
    log.setHost(request.getRemoteHost());
    log.setAddr(request.getRemoteAddr());
    log.setUser(request.getRemoteUser());
    log.setUrl(request.getRequestURL().toString());
    log.setUsuario((Usuario)userLogin);
    getLoginAuthorization().saveLog(log);
  }else{
    request.setAttribute(SECURITY_MSGLOGIN, WIDTHOUT_PERMISSIONS);
    request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
  }
  return;
}
else {
    String param_user = request.getParameter("j_u");
    String param_pass = request.getParameter("j_p");
    if (param_user != null) {
      LoginAuthorization authorization;
      authorization = getLoginAuthorization();
    try {
       UsuarioLogin userLogin;
       userLogin = authorization.getUsuario(param_user = param_user
                   .toUpperCase().trim());
    if (userLogin == null) {
```

Usuario no existe

```
       request.setAttribute(SECURITY_MSGLOGIN, "Usuario no registrado en el
         Sistema.");
       request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
     }
     else  {
       if (!userLogin.validaPassword(param_pass)) {
```

Clave invalida

```
     String requestURI =  request.getRequestURI();
     String contextPath=  request.getContextPath()+"/";
     int result = request.getRequestURI().indexOf("j_security_check");
     if (result != -1){
       request.getSession().setAttribute(SESSION_LAST_URI, contextPath);
     }else{
       request.getSession().setAttribute(SESSION_LAST_URI, requestURI);
}
  request.setAttribute(SECURITY_MSGLOGIN, "Contraseña no válida.");
  request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
}
else {
```

Clave correcta

```
String last_uri = (String)request.getSession().getAttribute(SESSION_LAST_URI);        uri =
request.getRequestURI().substring(request.getContextPath().length());
String logoff_uri = request.getContextPath()+LOG_OFF_URL;
```

```
request.getSession().setAttribute(SESSION_USERLOGIN, userLogin);
request.getSession().setAttribute(SESSION_LOGGED, Boolean.TRUE);
if (uri.indexOf("/j_security_check.html") != -1){
```

Verificar cambio de clave

```
Usuario usuario = (Usuario) userLogin;
Date fecha = usuario.getFechaCambioClave();
if (fecha == null){
  fecha = usuario.getFechaSolicitud();
}
Calendar calendar = Calendar.getInstance();
calendar.setTime(fecha);
calendar.set(Calendar.MONTH, calendar.get(Calendar.MONTH) + 3 );
if (Calendar.getInstance().compareTo(calendar) > 0 ){
  request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA, Usuario.INICIAL);
  }else
    if (usuario.getEstadoCambio().compareTo(Usuario.INICIAL) == 0){
    request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA, Usuario.INICIAL);
  }
response.sendRedirect(response.encodeRedirectURL("/"+request.getContextPath()+"/global.
do"));
   return;
  }
  if (last_uri != null && last_uri.compareTo(logoff_uri) != 0 && last_uri.compareTo
    (ACTION_URI) != 0) {
    if (!requestRequiresLoggedSession(last_uri.substring(request
        .getContextPath().length()), userLogin.getLoginName()) ){
```

Verificar cambio de clave

```
      Usuario usuario = (Usuario) userLogin;
      Date fecha = usuario.getFechaCambioClave();
      if (fecha == null){
      fecha = usuario.getFechaSolicitud();
    }
  Calendar calendar = Calendar.getInstance();
  calendar.setTime(fecha);
  calendar.set(Calendar.MONTH, calendar.get(Calendar.MONTH) + 3 );
  if (Calendar.getInstance().compareTo(calendar) > 0 ){
    request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA,
      Usuario.INICIAL);
  }else
     if (usuario.getEstadoCambio().compareTo(Usuario.INICIAL) == 0){
        request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA,
          Usuario.INICIAL);
     }
     request.getSession().removeAttribute(SESSION_LAST_URI);
     response.sendRedirect(response.encodeRedirectURL(last_uri));
     }else{
         request.setAttribute(SECURITY_MSGLOGIN, WIDTHOUT_PERMISSIONS);
         request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
         return;
  }
}else {
```

```
  if (!requestRequiresLoggedSession(uri, userLogin.getLoginName()) ) {

  Si este URL pertenece a usuario GUEST no problem
  Verificar cambio de clave
  Usuario usuario = (Usuario) userLogin;
  Date fecha = usuario.getFechaCambioClave();
  if (fecha == null){
    fecha = usuario.getFechaSolicitud();
  }
  Calendar calendar = Calendar.getInstance();
  calendar.setTime(fecha);
  calendar.set(Calendar.MONTH, calendar.get(Calendar.MONTH) + 3 );
  if (Calendar.getInstance().compareTo(calendar) > 0 ){
    request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA,
      Usuario.INICIAL);
}else
  if (usuario.getEstadoCambio().compareTo(Usuario.INICIAL) == 0){
   request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA, Usuario.INICIAL);
   }
   chain.doFilter(request, response);
   }else{
    request.setAttribute(SECURITY_MSGLOGIN, WIDTHOUT_PERMISSIONS);
    request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
   }
  }
 }
}
} catch (Throwable e) {  Error al traer información de la BD
  e.printStackTrace();
  request.setAttribute(SECURITY_MSGLOGIN, e.getMessage());
  request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
 }
}
 else {
 request.getSession().setAttribute(SESSION_LAST_URI, request.getRequestURI());
 request.getRequestDispatcher(LOG_IN_URL).forward(request, response);
 }
}
}
private boolean matchURL(String credencial, String uri) {
  try {
    URLPattern urlPattern = urlPatternFactory.createURLPattern(credencial, 0);
    URLPatternMatcher matcher = new URLPatternMatcher();
    return matcher.match(uri, urlPattern);
  } catch (Exception e) {
   }
return false;
}

Valida si el patron pertenece al usuario "INVITADO"
private boolean requestRequiresLoggedSession(String uri, String usrLogin) {
  if (uri.compareTo(ROOT_URI) == 0 || uri.compareTo(ACTION_URI) == 0){
    return false;
```

```
  }else{
  Iterator opciones = getLoginAuthorization().getOpciones(usrLogin).iterator();
  while (opciones.hasNext()) {
    CredencialAcceso credencial = (CredencialAcceso) opciones.next();
      if (matchURL(credencial.getPattern(), uri)) {
           return false;
         }
      }
    }
    return true;
 }

public LoginAuthorization getLoginAuthorization() {
  ApplicationContext springAppContext =                    WebApplicationContextUtils
     .getWebApplicationContext(filterConfig.getServletContext());
                   return
     (LoginAuthorization) springAppContext.getBean(loginAuthorizationParam);
}

public void destroy() {
  this.filterConfig = null;
}
}
```

Mensajes.java

```
package com.bvg.comun.web.ui;
public class Mensajes{
```

Mensajes Generales

```
public static final String APP_MENSAJE_GUARDADO = "app.mensaje.guardado";
public static final String APP_MENSAJE_BORRADO =  "app.mensaje.borrado";
public static final String APP_MENSAJE_NO_EXISTE =  "app.mensaje.no_existe";
public static final String APP_MENSAJE_NOMBRES_NAVES_NO_COINCIDEN =
  "app.mensaje.nombres_no_coinciden";
public static final String APP_MENSAJE_TRANSACCION_EXITOSA =
  "app.mensaje.transaccion_exitosa";
public static final String APP_MENSAJE_NO_USERLOGIN =
  "app.mensaje.no_userlogin";
public static final String APP_MENSAJE_CAMBIO_NOMBRE_NO_AUTORIZADO =
  "app.mensaje.cambio_nombre_no_autorizado";
public static final String
  APP_MENSAJE_NOMBRES_NAVES_NO_NOMBRE_ANTERIOR =
  "app.mensaje.cambio_nombre_no_nombre_anterior";
```

Mensajes de Error

```
public static final String APP_MENSAJE_ERROR = "app.mensaje.error";
public static final String APP_MENSAJE_ERROR_EXCEPCION=
  "app.mensaje.error.excepcion";
```

Botones

```
public static final String WEB_UI_BOTON_CREAR = "web.ui.boton.crear";
public static final String WEB_UI_BOTON_CONSULTAR = "web.ui.boton.consultar";
```

```
public static final String WEB_UI_BOTON_EDITAR = "web.ui.boton.editar";
public static final String WEB_UI_BOTON_GRABAR = "web.ui.boton.grabar";
public static final String WEB_UI_BOTON_CANCELAR = "web.ui.boton.cancelar";
public static final String WEB_UI_BOTON_ELIMINAR =  "web.ui.boton.eliminar";
public static final String WEB_UI_BOTON_BUSCAR = "web.ui.boton.buscar";
public static final String WEB_UI_BOTON_BUSQUEDA_AVANZADA =
  "web.ui.boton.busqueda_avanzada";

public Mensajes() {
  super();
}
}
```

GlobalAction.java

```
package com.bvg.comun.web;
import java.lang.reflect.Method;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.actions.DispatchAction;
import com.bvg.comun.persistencia.Eliminable;
import com.bvg.comun.security.model.auth.Menu;
import com.bvg.comun.security.model.auth.Rol;
import com.bvg.comun.security.model.auth.Usuario;
import com.bvg.comun.security.util.UsuarioLogin;
import com.bvg.comun.web.security.SecurityFilter;
import com.bvg.security.service.ModelManager;

public class GlobalAction extends DispatchAction{
```

Deficiones de atributos para generar acciones dinamicas en las páginas jsp's

```
public static final String ACCION = "ACCION";
public static final String METODO = "METODO";
public static final String PARAMETROS = "PARAMETROS";
```

URL de la acción de struts

```
public static final String GLOBAL_ACCION = "global";
```

Nombre del parametro de URL con el método a ejecutar en segunda instancia

```
public static String EJECUTAR_METODO = "ejecutar_metodo";
protected ModelManager manager = null;
```

Página de Inicio

```
public ActionForward index(ActionMapping mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response)
     throws Exception {
  UsuarioLogin usuarioLogin = (UsuarioLogin)
```

```
request.getSession().getAttribute(SecurityFilter.SESSION_USERLOGIN);
if (usuarioLogin != null){
  Usuario ususario =
  manager.findUsuarioByIdWhithRoles(usuarioLogin.getLoginName());
  request.getSession().setAttribute("usuario", ususario);
  StringBuffer menuWeb = new StringBuffer("");
  int cont=0;
  for (Iterator iter = ususario.getRoles().iterator(); iter.hasNext();) {
    Rol rol = (Rol) iter.next();
    List menus = new ArrayList(rol.getMenus());
    Collections.sort(menus);
    for (Iterator iterator = menus.iterator(); iterator.hasNext();) {
      Menu menu = (Menu) iterator.next();
        if (menu.getIdPadre() == null &&
          menu.getEstado().compareTo(Eliminable.ESTADO_ACTIVO) == 0 &&
          menu.isMenuMostrable()){
          menuWeb.append("this.item"+cont+"='"+menu.getNombre()+"';");
          menuWeb.append(
```

Fondo y borde grupo

```
           "this.item_background_color"+cont+" = '#ffffff';" +
           "this.item_border_color"+cont+" = '#9b9999';" +
           "this.item_border_color"+cont+" = '#9b9999';" +
           "this.item_border_color_roll"+cont+" = '#9b9999';" +
           "this.item_border_color_roll"+cont+" = '#9b9999';" +
           "this.item_border_width"+cont+" = '0';" +
```

Fuente grupo

```
           "this.item_background_color_roll"+cont+" = 'transparent';" +
           "this.item_font_family"+cont+" = 'Verdana';" +
           "this.item_font_weight"+cont+" = 'bold';" +
           "this.item_font_style"+cont+" = 'normal';" +
           "this.item_font_size"+cont+" = '11px';" +
           "this.item_padding"+cont+" = '3,5,3,3';"
           );
           cont++;
           List submenus = new ArrayList();
          for (Iterator iter2 = rol.getMenus().iterator(); iter2.hasNext();) {
            Menu subMenu = (Menu) iter2.next();
            if (subMenu.getIdPadre() != null &&
             subMenu.getIdPadre().compareTo(menu.getId()) == 0){
               submenus.add(subMenu);
            }
          }
          Collections.sort(submenus);
          for (Iterator iter2 = submenus.iterator(); iter2.hasNext();) {
            Menu subMenu = (Menu) iter2.next();
            if (subMenu.isMenuMostrable()) {
             menuWeb.append("this.item"+cont+"='
             "+subMenu.getNombre()+"';");
          if (subMenu.getUrl() != null && subMenu.getUrl().length()>0 &&
            subMenu.getEstado().compareTo(Eliminable.ESTADO_ACTIVO) == 0) {
```

```
                  menuWeb.append("this.url"+cont+" =
                   \"../../.."+subMenu.getUrl()+"\";" +
                      "this.url_target"+cont+" = 'iFrameAdmin';");
              }
             cont++;
          }
        }
```

Divisor de Grupos

```
      menuWeb.append("this.item"+cont+"=' ';");
      menuWeb.append("/* Línea divisora */" +
        "this.item_font_family"+cont+" = 'Arial';" +
        "this.item_font_weight"+cont+" = 'normal';" +
        "this.item_font_style"+cont+" = 'normal';" +
        "this.item_font_size"+cont+" = '0px';" +
        "this.item_border_width"+cont+" = '0';"
        );
        cont++;
    }
  }
}
request.setAttribute("menu",menuWeb);
}else{
  return mapping.findForward("login");
}
  return mapping.findForward("index");
}

public ActionForward bienvenida(ActionMapping mapping, ActionForm form,
  HttpServletRequest request, HttpServletResponse response)   throws Exception{
  String isCambio = (String)
  request.getSession().getAttribute(Usuario.CLAVE_CAMBIADA);
  if (isCambio != null && isCambio.compareTo(Usuario.INICIAL) == 0) {
    UsuarioLogin usuarioLogin = (UsuarioLogin)
    request.getSession().getAttribute(SecurityFilter.SESSION_USERLOGIN);
   if (usuarioLogin != null) {
      request.getRequestDispatcher("/security.do?method=cambioClave")
      .forward(request, response);
    }
  }
  return mapping.findForward("paginaEnBlanco");
 }
```

Página Vacía

```
public ActionForward paginaVacia(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response)
     throws Exception{
  return mapping.findForward("paginaMensajes");
}
```

Acción no especificada

```
public ActionForward unspecified(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response) throws Exception {
```

```
  return index(mapping, form, request, response);
}

public void setModelManager(ModelManager mgr) {
  this.manager = mgr;
 }
```

Métodos Útiles

```
public boolean isMetodoCorrecto(String metodo) {
  Method[] metodos = this.getClass().getMethods();
  if (metodos != null && metodos.length > 0 ) {
    Method method = null;
    for (int i = 0; i < metodos.length; i++) {
      method = metodos[i];
      if (method.getName().compareTo(metodo.trim()) == 0 ) {
         return true;
      }
    }
   }
   return false;
 }
}
```

Manager.java

```
package com.bvg.security.service;
import com.bvg.comun.service.ComunManager;

public class Manager extends ComunManager implements ModelManager{
   public Manager() {
         super();
   }
}
```

Modelmanager.java

```
package com.bvg.security.service;
import com.bvg.comun.service.ComunModelManager;

public interface ModelManager extends ComunModelManager {
}
```

SecurityAction.java

Tiene los metodos del struts

```
package com.bvg.security.web;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collections;
import java.util.HashMap;
import java.util.Iterator;
```

```
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionMessages;
import org.apache.struts.action.DynaActionForm;
import com.bvg.comun.security.model.auth.Menu;
import com.bvg.comun.security.model.auth.Rol;
import com.bvg.comun.security.model.auth.Usuario;
import com.bvg.comun.security.util.SHA1BASE64;
import com.bvg.comun.security.util.UsuarioLogin;
import com.bvg.comun.web.GlobalAction;
import com.bvg.comun.web.security.SecurityFilter;
import com.bvg.comun.web.ui.Mensajes;

public class SecurityAction extends GlobalAction{
```

URL de la acción de struts

```
  public static final String SECURITY_ACCION = "security";
   public ActionForward listarUsuarios(ActionMapping mapping, ActionForm
      form, HttpServletRequest request, HttpServletResponse response)
      throws Exception {
      String accion = (String) request.getSession().getAttribute(ACCION);
      if (accion == null || accion.length() <= 0) {
         request.getSession().setAttribute(ACCION, SECURITY_ACCION);
      }
      DynaActionForm dynaForm = (DynaActionForm) form;
      String textToFind = (String) dynaForm.get("textToFind");
      String cmbTextToFind = (String) dynaForm.get("cmbTextToFind");
      String cmbEstado = (String) dynaForm.get("cmbEstado");
      List usuarios = null;
      HashMap filtros = new HashMap();
      List result = new ArrayList();
      if (cmbTextToFind != null && cmbTextToFind.length() > 0 ) {
         if (cmbTextToFind.compareTo("NOMBRE") == 0 && textToFind != null
            && textToFind.length() > 0) {
            filtros.put("url like ?",textToFind.toUpperCase().trim()+"%");
              filtros.put("nombre like ?",textToFind.toUpperCase().trim()+"%");
         }
         if (cmbTextToFind.compareTo("USUARIO") == 0 && textToFind != null
            && textToFind.length() > 0) {
           filtros.put("id like ?",textToFind+"%");
        }
 if(cmbEstado != null && cmbEstado.length() >0 && cmbEstado.compareTo("TODOS") != 0) {
            filtros.put("estado = ?",cmbEstado);
}
            usuarios = manager.getUsuarios(filtros);
            for (Iterator iter = usuarios.iterator(); iter.hasNext();) {
               Usuario usuario = (Usuario) iter.next();
               boolean sw = false;
```

```
          if (cmbTextToFind.compareTo("NOMBRE") == 0 && textToFind !=
           null && textToFind.length() > 0) {
            if (usuario.getNombre().toUpperCase().trim()
             .indexOf(textToFind.toUpperCase().trim()) == -1){
               sw = true;
             }
         }
         if (cmbTextToFind.compareTo("USUARIO") == 0 && textToFind !=
         null && textToFind.length() > 0) {
             if (usuario.getId().toUpperCase().trim()
              .indexOf(textToFind.toUpperCase().trim()) == -1) {
                   sw = true;
             }
         }
         if (cmbEstado != null && cmbEstado.length() > 0 &&
          cmbEstado.compareTo("TODOS") != 0){
            if (usuario.getEstado().toUpperCase().trim()
              .indexOf(cmbEstado.toUpperCase().trim()) == -1){
                   sw = true;
             }
         }
         if (!sw){
            result.add(usuario);
         }
       }
    }
    request.setAttribute("items", result);
    return mapping.findForward("listarUsuarios");
}

public ActionForward editarUsuario(ActionMapping mapping, ActionForm
   form, HttpServletRequest request, HttpServletResponse response)
   throws Exception {
  String accion = (String) request.getSession().getAttribute(ACCION);
  if (accion == null || accion.length() <= 0)  {
    request.getSession().setAttribute(ACCION, SECURITY_ACCION);
  }
  String id_usuario = request.getParameter("id_usuario");
  Usuario usuario = null;
  if (id_usuario != null && id_usuario.length() > 0) {
       usuario = manager.findUsuarioById(id_usuario);
       usuario.setConfirmClave(usuario.getClave());
       }else{
          usuario = new Usuario();
       }
       request.setAttribute("usuario", usuario);
       DynaActionForm dynaForm = (DynaActionForm) form;
       dynaForm.set("usuario", usuario);
       return mapping.findForward("editarUsuario");
   }

   public ActionForward cambioClave(ActionMapping mapping, ActionForm
      form, HttpServletRequest request, HttpServletResponse response)
```

```
      throws Exception {
        String accion = (String) request.getSession().getAttribute(ACCION);
        if (accion == null || accion.length() <= 0) {
            request.getSession().setAttribute(ACCION, SECURITY_ACCION);
         }
        UsuarioLogin usuarioLogin = (UsuarioLogin)
           request.getSession().getAttribute
           (SecurityFilter.SESSION_USERLOGIN);
         ActionMessages msg = new ActionErrors();
         if (usuarioLogin != null) {
             String isCambio = (String)
           request.getSession().getAttribute(Usuario.CLAVE_CAMBIADA);
             Usuario usuario =
                  manager.findUsuarioById(usuarioLogin.getLoginName());
              if (isCambio == null) {
                    request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA,
                     Usuario.INICIAL);
                    }else{
                        usuario.setEstadoCambio(Usuario.INICIAL);
                        manager.guardarObjeto(usuario);
                    }
                    request.setAttribute("usuario", usuario);
                    DynaActionForm dynaForm = (DynaActionForm) form;
                          dynaForm.set("usuario", usuario);
                    return mapping.findForward("cambioClave");
              }
              msg.add(ActionMessages.GLOBAL_MESSAGE, new
            ActionMessage(Mensajes.APP_MENSAJE_ERROR, "No esta
                accediendo a esta página de manera correcta."));
                    saveMessages(request, msg);
              return paginaVacia(mapping,form,request, response);
  }

public ActionForward olvidoClave(ActionMapping mapping, ActionForm
     form, HttpServletRequest request, HttpServletResponse response)
     throws Exception  {
    String accion = (String) request.getSession().getAttribute(ACCION);
    if (accion == null || accion.length() <= 0) {
        request.getSession().setAttribute(ACCION, SECURITY_ACCION);
    }
    ActionMessages msg = new ActionErrors();
    String isOlvido = request.getParameter("isOlvido");
    if (isOlvido != null &&
       isOlvido.compareTo(request.getSession().getId())== 0) {
         String param_user = request.getParameter("j_u");
         if (param_user != null && param_user.length() > 0){
           String id_usuario = request.getParameter("id_usuario");
           Usuario usuario =
              manager.findUsuarioById(param_user.trim().toUpperCase());
          if (usuario != null) {
           if (id_usuario != null && id_usuario.length() > 0) {
               String respuesta = request.getParameter("usuario.respuesta");
               if (respuesta != null &&
```

```
                respuesta.trim().toUpperCase().compareTo
                (usuario.getRespuesta().trim().toUpperCase()) == 0) {
                  request.getSession().setAttribute
                   (SecurityFilter.SESSION_USERLOGIN, usuario);
                 request.getSession().setAttribute
                   (Usuario.CLAVE_CAMBIADA, Usuario.INICIAL);
                  return cambioClave(mapping, form, request,response);
                  }else{
                     msg.add(ActionMessages.GLOBAL_MESSAGE, new
                 ActionMessage(Mensajes.APP_MENSAJE_ERROR, "No es la
                    respuesta de la pregunta."));
                  }
           }
                  request.setAttribute("usuario", usuario);
                  DynaActionForm dynaForm = (DynaActionForm) form;
                  dynaForm.set("usuario", usuario);
                  if (usuario.getRespuesta() == null ||
                   usuario.getRespuesta().length() <= 0 ) {
                      msg.add(ActionMessages.GLOBAL_MESSAGE, new
                  ActionMessage(Mensajes.APP_MENSAJE_ERROR, "El
                        usuario no tiene una respueta asignado a la pregunta."));
                              saveMessages(request, msg);
                              return mapping.findForward("logoff");
                     }
                     saveMessages(request, msg);
                     return mapping.findForward("olvidoClave");
                  }
           }
        }
        msg.add(ActionMessages.GLOBAL_MESSAGE, new
       ActionMessage(Mensajes.APP_MENSAJE_ERROR, "No esta
            accediendo a esta página de manera correcta."));
        saveMessages(request, msg);
        return paginaVacia(mapping,form,request, response);
    }

public ActionForward agregarRol(ActionMapping mapping, ActionForm
    form, HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    String accion = (String) request.getSession().getAttribute(ACCION);
    if (accion == null || accion.length() <= 0) {
        request.getSession().setAttribute(ACCION, SECURITY_ACCION);
    }
    String id_usuario = request.getParameter("id_usuario");
    ActionMessages msg = new ActionErrors();
      if (id_usuario != null && id_usuario.length() > 0) {
         String id_rol = request.getParameter("id_rol");
         if (id_rol != null && id_rol.length() > 0){
            Usuario usuario = manager.findUsuarioById(id_usuario);
            Rol rol = null;
            for (Iterator iter = manager.getRoles(null).iterator();
                 iter.hasNext();){
                 Rol rol2 = (Rol) iter.next();
```

```
                    if (rol2.getId().compareTo(Long.valueOf(id_rol)) == 0) {
                       rol = rol2;
                    break;
                    }
              }
              for (Iterator iter = usuario.getRoles().iterator(); iter.hasNext();) {
                   Rol rol2 = (Rol) iter.next();
                   if (rol2.getId().compareTo(Long.valueOf(id_rol)) == 0) {
                      rol = null;
                      break;
                   }
              }
              if(rol != null) {
                    usuario.getRoles().add(rol);
                    manager.guardarObjeto(usuario);
                    msg.add(ActionMessages.GLOBAL_MESSAGE, new
                       ActionMessage(Mensajes.APP_MENSAJE_
                        TRANSACCION_EXITOSA, usuario.getNombre()+":
                        "+rol.getNombre()));
                    }else{
                      msg.add(ActionMessages.GLOBAL_MESSAGE, new
                     ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                    "El rol ya ha sido asignado a este usuario anteriormente."));
                   }
                   }else{
                      msg.add(ActionMessages.GLOBAL_MESSAGE, new
                      ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                      "Escoja un rol para asignarlo al usuario."));
                   }
                   }else{
                     msg.add(ActionMessages.GLOBAL_MESSAGE, new
                 ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                 "Escoja un usuario para agregar un rol."));
                   }
                   saveMessages(request, msg);
                   return editarUsuario(mapping,form,request,response);
}

public ActionForward quitarRol(ActionMapping mapping, ActionForm form,
   HttpServletRequest request, HttpServletResponse response) throws Exception {
  String accion = (String) request.getSession().getAttribute(ACCION);
   if (accion == null || accion.length() <= 0) {
    request.getSession().setAttribute(ACCION, SECURITY_ACCION);
 }
 String id_usuario = request.getParameter("id_usuario");
 ActionMessages msg = new ActionErrors();
 if (id_usuario != null && id_usuario.length() > 0) {
  String id_rol = request.getParameter("id_rol");
  if (id_rol != null && id_rol.length() > 0) {
     Usuario usuario = manager.findUsuarioById(id_usuario);
     Rol rol = null;
     for (Iterator iter = usuario.getRoles().iterator(); iter.hasNext();){
        Rol rol2 = (Rol) iter.next();
```

```
            if (rol2.getId().compareTo(Long.valueOf(id_rol)) == 0) {
            rol = rol2;
            break;
          }
    }
    if(rol != null && usuario.getRoles().contains(rol)) {
      usuario.getRoles().remove(rol);
        manager.guardarObjeto(usuario);
        msg.add(ActionMessages.GLOBAL_MESSAGE, new
    ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION
    _EXITOSA, usuario.getNombre()+": "+rol.getNombre()));
  }
  }else{
       msg.add(ActionMessages.GLOBAL_MESSAGE, new
      ActionMessage(Mensajes.APP_MENSAJE_ERROR,
      "Escoja un rol para asignarlo al usuario."));
  }
  }else{
       msg.add(ActionMessages.GLOBAL_MESSAGE, new
      ActionMessage(Mensajes.APP_MENSAJE_ERROR,
      "Escoja un usuario para agregar un rol."));
  }
  saveMessages(request, msg);
  return editarUsuario(mapping,form,request,response);
}

public ActionForward guardarUsuario(ActionMapping mapping, ActionForm
    form, HttpServletRequest request, HttpServletResponse response)
    throws Exception {
   String accion = (String) request.getSession().getAttribute(ACCION);
   if (accion == null || accion.length() <= 0) {
        request.getSession().setAttribute(ACCION,
      SECURITY_ACCION);
   }
   DynaActionForm usuarioForm = (DynaActionForm) form;
   Usuario usuario = (Usuario) usuarioForm.get("usuario");
   if (usuario.getFechaSolicitud() == null) {
       usuario.setFechaSolicitud(Calendar.getInstance().getTime());
   }
  ActionMessages msg = new ActionErrors();
  if (usuario.getClave() != null && usuario.getConfirmClave() != null &&
     usuario.getClave().compareTo(usuario.getConfirmClave()) == 0){
       boolean sw = false;
       String id_usuario = request.getParameter("id_usuario");
       if (id_usuario != null && id_usuario.length() > 0){
           sw = false;
       }else{
      Usuario usurio2 = manager.findUsuarioById(usuario.getId());
       if (usurio2 != null) {
          sw = true;
       }
    } if (!sw) {
      usuario.setId(usuario.getId().toUpperCase());
```

```
            if (usuario.getEstadoCambio() == null){
                usuario.setEstadoCambio(Usuario.INICIAL);
            }
            String isCambio = (String)
           request.getSession().getAttribute(Usuario.CLAVE_CAMBIADA);
           String clave_anterior = request.getParameter("clave_anterior");
           if (clave_anterior != null &&
               clave_anterior.compareTo(usuario.getClave())== 0) {
               //ignore
           }else{
           usuario.setClave(SHA1BASE64.encriptar(usuario.getClave()));
        }  if (isCambio != null && isCambio.compareTo(Usuario.INICIAL) == 0){
           usuario.setEstadoCambio(Usuario.CLAVE_CAMBIADA);
           usuario.setFechaCambioClave(Calendar.getInstance().getTime());
           manager.guardarObjeto(usuario);
           request.getSession().setAttribute(Usuario.CLAVE_CAMBIADA,
                Usuario.CLAVE_CAMBIADA);
           msg.add(ActionMessages.GLOBAL_MESSAGE, new
           ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION_EXITOSA,
           usuario.getNombre()));  return mapping.findForward("logoff");
           }else{
           manager.guardarObjeto(usuario);
           msg.add(ActionMessages.GLOBAL_MESSAGE, new
           ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION_EXITOSA,
           usuario.getNombre()));
        }
        }else{
        msg.add(ActionMessages.GLOBAL_MESSAGE, new
        ActionMessage(Mensajes.APP_MENSAJE_ERROR,"El nombre de usuario: " +
        usuario.getId() + " ya se encuentra asignado.   Por favor ingrese otro."));
    }
    }else{
    msg.add(ActionMessages.GLOBAL_MESSAGE, new
    ActionMessage(Mensajes.APP_MENSAJE_ERROR,
    "La clave y su confirmación tienen que ser iguales."));
  }
  saveMessages(request, msg);
  request.setAttribute("usuario", usuario);
  String isCambio = (String)
  request.getSession().getAttribute(Usuario.CLAVE_CAMBIADA);
  if (isCambio != null && isCambio.compareTo(Usuario.INICIAL) == 0) {
     return mapping.findForward("cambioClave");
  }
  return mapping.findForward("editarUsuario");
}

public ActionForward listarRoles(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    String accion = (String) request.getSession().getAttribute(ACCION);
    if (accion == null || accion.length() <= 0) {
        request.getSession().setAttribute(ACCION, SECURITY_ACCION);
    }
```

```
 String id_usuario = request.getParameter("id_usuario");
 System.out.println(id_usuario);
 DynaActionForm dynaForm = (DynaActionForm) form;
 String textToFind = (String) dynaForm.get("textToFind");
 String cmbTextToFind = (String) dynaForm.get("cmbTextToFind");
 String cmbEstado = (String) dynaForm.get("cmbEstado");
 List items = null;
 HashMap filtros = new HashMap();
 List result = new ArrayList();
if (cmbTextToFind != null && cmbTextToFind.length() > 0 ) {
    if (cmbTextToFind.compareTo("NOMBRE") == 0 && textToFind != null
       && textToFind.length() > 0) {
         filtros.put("nombre like ?",textToFind.toUpperCase().trim()+"%");
    }
    if (cmbTextToFind.compareTo("DESCRIPCION") == 0 && textToFind !=
        null && textToFind.length() > 0) {
         filtros.put("descripcion like ?",textToFind+"%");
    }
    if (cmbEstado != null && cmbEstado.length() > 0 &&
        cmbEstado.compareTo("TODOS") != 0) {
        filtros.put("estado = ?",cmbEstado);
    }
    items = manager.getRoles(filtros);
    for (Iterator iter = items.iterator(); iter.hasNext();){
        Rol rol = (Rol) iter.next();
        boolean sw = false;
        if (cmbTextToFind.compareTo("NOMBRE") == 0 && textToFind != null
         && textToFind.length() > 0) {
           if (rol.getNombre().toUpperCase().trim().indexOf
               (textToFind.toUpperCase().trim()) == -1) {
               sw = true;
           }
         }
         if (cmbTextToFind.compareTo("DESCRIPCION") == 0 && textToFind
           != null && textToFind.length() > 0) {
            if (rol.getDescripcion().toUpperCase().trim()
              .indexOf(textToFind.toUpperCase().trim()) == -1) {
                 sw = true;
            }
         }
       if (cmbEstado != null && cmbEstado.length() > 0 &&
           cmbEstado.compareTo("TODOS") != 0) {
             if (rol.getEstado().toUpperCase().trim().indexOf
              (cmbEstado.toUpperCase().trim()) == -1) {
                  sw = true;
              }
           }
           if (!sw) {
                  result.add(rol);
           }
         }
     }
     request.setAttribute("items", items);
```

```
    return mapping.findForward("listarRoles");
}

public ActionForward editarRol(ActionMapping mapping, ActionForm form,
     HttpServletRequest request, HttpServletResponse response)
     throws Exception {
   String accion = (String) request.getSession().getAttribute(ACCION);
   if (accion == null || accion.length() <= 0) {
     request.getSession().setAttribute(ACCION, SECURITY_ACCION);
   }
   String id = request.getParameter("id_rol");
   Rol entidad = null;
   if (id != null && id.length() > 0) {
        entidad = manager.findRolWithMenus(id);
   }else{
       entidad = new Rol();
   }
   DynaActionForm dynaForm = (DynaActionForm) form;
   dynaForm.set("rol", entidad);
   request.setAttribute("rol", entidad);

   List menus = new ArrayList();
   for (Iterator iter = entidad.getMenus().iterator(); iter.hasNext();){
   Menu menu = (Menu) iter.next();
   if (menu.getIdPadre() == null) {
      menus.add(menu);
       }
  }
  request.setAttribute("menus", menus);
  return mapping.findForward("editarRol");
}

public ActionForward agregarMenu(ActionMapping mapping, ActionForm
     form, HttpServletRequest request, HttpServletResponse response)
     throws Exception {
String accion = (String) request.getSession().getAttribute(ACCION);
if (accion == null || accion.length() <= 0) {
      request.getSession().setAttribute(ACCION, SECURITY_ACCION);
}
String id_rol = request.getParameter("id_rol");
ActionMessages msg = new ActionErrors();
if (id_rol != null && id_rol.length() > 0) {
  String id_menu = request.getParameter("id_menu");
  if (id_menu != null && id_menu.length() > 0) {
      Rol rol = manager.findRolWithMenus(id_rol);
      Menu menu = null;
      for (Iterator iter = manager.getMenus(null).iterator(); iter.hasNext();){
          Menu menu2 = (Menu) iter.next();
          if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0) {
             menu = menu2;
             break;
          }
      }
```

```
for (Iterator iter = rol.getMenus().iterator(); iter.hasNext();){
     Menu menu2 = (Menu) iter.next();
     System.out.println(menu2.getNombre());
     if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0) {
         menu = null;
         break;
 }
 boolean sw = false;
 for (Iterator iterator = menu2.getMenus().iterator(); iterator.hasNext();){
     Menu menu3 = (Menu) iterator.next();
     if (menu3.getId().compareTo(Long.valueOf(id_menu)) == 0) {
         menu = menu3;
         sw = true;
         break;
      }
  }
  if(sw){
       break;
  }
}
if(menu != null){
      rol.getMenus().add(menu);
      System.out.println(menu.getNombre());
      boolean sw = false;
      for (Iterator iter = menu.getMenus().iterator(); iter.hasNext();){
          Menu menu2 = (Menu) iter.next();
          System.out.println(menu2.getNombre());
          sw = false;
          for (Iterator iterator = rol.getMenus().iterator(); iterator.hasNext();){
                Menu menu3 = (Menu) iterator.next();
                System.out.println(menu3.getNombre());
                if(menu2.getId().compareTo(menu3.getId()) == 0) {
                    sw = true;
                    break;
                }
           }
           if(!sw) {
                rol.getMenus().add(menu2);
           }
      }
manager.guardarObjeto(rol);
msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
(Mensajes.APP_MENSAJE_TRANSACCION_EXITOSA,
 rol.getNombre()+": "+menu.getNombre()));
 }else{
        msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
       (Mensajes.APP_MENSAJE_ERROR,  "El menu ya ha sido asignado a este rol
       anteriormente."));
    }
   }else{
       msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
       (Mensajes.APP_MENSAJE_ERROR, "Escoja un menu para asignarlo al rol."));
 }
```

```
      }else{
              msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
              (Mensajes.APP_MENSAJE_ERROR,  "Escoja un rol para agregar un menu."));
      }
     saveMessages(request, msg);
     return editarRol(mapping,form,request,response);
}

public ActionForward quitarMenu(ActionMapping mapping, ActionForm form, HttpServlet
  Requestrequest, HttpServletResponse response)  throws Exception {
     String accion = (String) request.getSession().getAttribute(ACCION);;
     if (accion == null || accion.length() <= 0) {
        request.getSession().setAttribute(ACCION, SECURITY_ACCION);
     }
    String id_rol = request.getParameter("id_rol");
    ActionMessages msg = new ActionErrors();
    if (id_rol != null && id_rol.length() > 0) {
        String id_menu = request.getParameter("id_menu");
        if (id_menu != null && id_menu.length() > 0) {
        Rol rol = manager.findRolWithMenus(id_rol);
          Menu menu = null;
          for (Iterator iter = rol.getMenus().iterator(); iter.hasNext();){
             Menu menu2 = (Menu) iter.next();
             if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0){
                  menu = menu2;
                  break;
            }
        }
        if(menu != null && rol.getMenus().contains(menu)){
          rol.getMenus().remove(menu);
          for (Iterator iter = menu.getMenus().iterator(); iter.hasNext();){
             Menu menu2 = (Menu) iter.next();
             Menu menu3 = null;
            for (Iterator iterator = rol.getMenus().iterator(); iterator.hasNext();){
                menu3 = (Menu) iterator.next();
                if(menu2.getId().compareTo(menu3.getId()) == 0) {
                    break;
                  }else{
                     menu3 = null;
                  }
                }
                if(menu3 != null && rol.getMenus().contains(menu3)) {
                   rol.getMenus().remove(menu2);
                }
            }
            manager.guardarObjeto(rol);
            msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
           (Mensajes.APP_MENSAJE_TRANSACCION_EXITOSA, rol.getNombre()+":
           "+menu.getNombre()));
           }
            }else{
                msg.add(ActionMessages.GLOBAL_MESSAGE, newActionMessage
               (Mensajes.APP_MENSAJE_ERROR, "Escoja un menu para asignarlo al rol."));
```

```
            }
        }else{
        msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
        (Mensajes.APP_MENSAJE_ERROR, "Escoja un rol para agregar un menu."));
    }
    saveMessages(request, msg);
    return editarRol(mapping,form,request,response);
}

public ActionForward quitarSubmenuRol(ActionMapping mapping,
  ActionForm form, HttpServletRequest request, HttpServletResponse response)  throws
  Exception {
   String accion = (String) request.getSession().getAttribute(ACCION);;
   if (accion == null || accion.length() <= 0) {
      request.getSession().setAttribute(ACCION, SECURITY_ACCION);
   }
   String id_rol = request.getParameter("id_rol");
   ActionMessages msg = new ActionErrors();
   if (id_rol != null && id_rol.length() > 0) {
      String id_menu = request.getParameter("id_sub_menu_rol");
      if (id_menu != null && id_menu.length() > 0) {
         Rol rol = manager.findRolWithMenus(id_rol);
         Menu menu = null;
         for (Iterator iter = rol.getMenus().iterator(); iter.hasNext();){
            Menu menu2 = (Menu) iter.next();
            if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0){
               menu = menu2;
               break;
            }
         }
         if(menu != null && rol.getMenus().contains(menu)) {
            rol.getMenus().remove(menu);
            for (Iterator iter = menu.getMenus().iterator(); iter.hasNext();){
               Menu menu2 = (Menu) iter.next();
            Menu menu3 = null;
               for (Iterator iterator = rol.getMenus().iterator(); iterator.hasNext();){
                  menu3 = (Menu) iterator.next();
                  if(menu2.getId().compareTo(menu3.getId()) == 0){
                     break;
                  }else{
                     menu3 = null;
               }
         }
         if(menu3 != null && rol.getMenus().contains(menu3)) {
            rol.getMenus().remove(menu2);
         }
    }
    manager.guardarObjeto(rol);
    msg.add(ActionMessages.GLOBAL_MESSAGE, new
    ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION_EXITOSA,
        rol.getNombre()+": "+menu.getNombre()));
    }
}else{
```

```
msg.add(ActionMessages.GLOBAL_MESSAGE, new
ActionMessage(Mensajes.APP_MENSAJE_ERROR, "Escoja un menu para asignarlo al
rol."));
}
}else{
    msg.add(ActionMessages.GLOBAL_MESSAGE, new ActionMessage
    (Mensajes.APP_MENSAJE_ERROR, "Escoja un rol para agregar un menu."));
}
saveMessages(request, msg);
return editarRol(mapping,form,request,response);
}

public ActionForward guardarRol(ActionMapping mapping, ActionForm form,
     HttpServletRequest request, HttpServletResponse response) throws
     Exception {
     String accion = (String) request.getSession().getAttribute(ACCION);
     if (accion == null || accion.length() <= 0) {
          request.getSession().setAttribute(ACCION, SECURITY_ACCION);
     }
     DynaActionForm dynaForm = (DynaActionForm) form;
     Rol entidad = (Rol) dynaForm.get("rol");
     ActionMessages msg = new ActionErrors();
     manager.guardarObjeto(entidad);
     msg.add(ActionMessages.GLOBAL_MESSAGE, new
     ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION_EXITOSA,
         entidad.getNombre()));
     saveMessages(request, msg);
     request.setAttribute("rol", entidad);
     List menus = new ArrayList();
          for (Iterator iter = entidad.getMenus().iterator(); iter.hasNext();){
             Menu menu = (Menu) iter.next();
             if (menu.getIdPadre() == null) {
                menus.add(menu);
             }
          }
           request.setAttribute("menus", menus);
           return mapping.findForward("editarRol");
}

public ActionForward listarMenus(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    String accion = (String) request.getSession().getAttribute(ACCION);
    if (accion == null || accion.length() <= 0) {
          request.getSession().setAttribute(ACCION, SECURITY_ACCION);
    }
    DynaActionForm dynaForm = (DynaActionForm) form;
    String textToFind = (String) dynaForm.get("textToFind");
    String cmbTextToFind = (String) dynaForm.get("cmbTextToFind");
    String cmbEstado = (String) dynaForm.get("cmbEstado");
    String cmbMostrable = (String) dynaForm.get("cmbMostrable");
    String cmbAgrupados = (String) dynaForm.get("cmbAgrupados");
```

```
List items = null;
HashMap filtros = new HashMap();
List result = new ArrayList();

if (cmbTextToFind != null && cmbTextToFind.length() > 0 ) {
      if (cmbTextToFind.compareTo("NOMBRE") == 0 && textToFind != null
        && textToFind.length() > 0) {
           filtros.put("nombre like ?",textToFind.toUpperCase().trim()+"%");
      }
      if (cmbTextToFind.compareTo("URL") == 0 && textToFind != null &&
        textToFind.length() > 0) {
           filtros.put("url like ?",textToFind.toUpperCase().trim()+"%");
      }
      if (cmbEstado != null && cmbEstado.length() > 0 &&
        cmbEstado.compareTo("TODOS") != 0) {
          filtros.put("estado = ?",cmbEstado);
      }
      if (cmbMostrable != null && cmbMostrable.length() > 0 ) {
          filtros.put("mostrable = ?",cmbMostrable);
    }else{
           filtros.put("mostrable = ?","S");
      }
      items = manager.getMenus(filtros);
    for (Iterator iter = items.iterator(); iter.hasNext();){
          Menu menu = (Menu) iter.next();
          boolean sw = false;
          if (cmbTextToFind.compareTo("NOMBRE") == 0 && textToFind !=
            null && textToFind.length() > 0) {
              if (menu.getNombre().toUpperCase().trim().indexOf
               (textToFind.toUpperCase().trim()) == -1) {
                    sw = true;
               }
          }
         if (cmbTextToFind.compareTo("URL") == 0 && textToFind != null &&
          textToFind.length() > 0) {
          if (menu.getUrl().toUpperCase().trim().indexOf
              (textToFind.toUpperCase().trim()) == -1) {
                  sw = true;
            }
         }
         if (cmbEstado != null && cmbEstado.length() > 0 &&
          cmbEstado.compareTo("TODOS") != 0) {
             if (menu.getEstado().indexOf(cmbEstado) == -1) {
                     sw = true;
                  }
               }
               if (cmbMostrable != null && cmbMostrable.length() > 0 ) {
                     if (menu.getMostrable().indexOf(cmbMostrable) == -1) {
                        sw = true;
                     }
                     }else{
                        if (menu.getMostrable().indexOf("S") == -1) {
                             sw = true;
```

```
                    }
                }
                if (cmbAgrupados != null && cmbAgrupados.length() > 0 &&
                 cmbAgrupados.compareTo("AGRUPADOS")== 0) {
                  if (menu != null && menu.getIdPadre() != null) {
                sw = true;
                    }
                }
                if(!sw) {
                    result.add(menu);
                }
            }
            Collections.sort(result);
        }
        request.setAttribute("items", result);
        return mapping.findForward("listarMenus");
}

public ActionForward editarMenu(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception{
    String accion = (String) request.getSession().getAttribute(ACCION);
     if (accion == null || accion.length() <= 0) {
           request.getSession().setAttribute(ACCION, SECURITY_ACCION);
     }
     String id = request.getParameter("id_menu");
     Menu entidad = null;
     if (id != null && id.length() > 0) {
           entidad = manager.findMenuById(id);
     }else{
           entidad = new Menu();
     }
     DynaActionForm dynaForm = (DynaActionForm) form;
     dynaForm.set("menu", entidad);
     request.setAttribute("menu", entidad);
     return mapping.findForward("editarMenu");
}

public ActionForward agregarSubMenu(ActionMapping mapping, ActionForm
    form, HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    String accion = (String) request.getSession().getAttribute(ACCION);
     if (accion == null || accion.length() <= 0){
         request.getSession().setAttribute(ACCION, SECURITY_ACCION);
      }
       String id_menu_padre = request.getParameter("id_menu_padre");
       ActionMessages msg = new ActionErrors();
       if (id_menu_padre != null && id_menu_padre.length() > 0) {
            String id_menu = request.getParameter("id_menu");
            if (id_menu != null && id_menu.length() > 0) {
                Menu menuPadre = manager.findMenuById(id_menu_padre);
                if (menuPadre!= null && menuPadre.getId().compareTo
                 (Long.valueOf(id_menu)) != 0) {
```

```
                Menu menu = null;
                for (Iterator iter = manager.getMenus(null).iterator();
                  iter.hasNext();) {
                     Menu menu2 = (Menu) iter.next();
                     if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0) {
                        menu = menu2;
                        break;
                     }
                }
              for (Iterator iter = menuPadre.getMenus().iterator(); iter.hasNext();){
                   Menu menu2 = (Menu) iter.next();
                   if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0) {
                      menu = null;
                      break;
                   }
               }
               if(menu != null){
                   if (menu.getMenus() != null && menu.getMenus().size() > 0){
                      msg.add(ActionMessages.GLOBAL_MESSAGE, new
                   ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION
                      _EXITOSA, "Sólo se permite un nivel de menus.  El menu
                      que esta agregando tiene asignados submenus."));
                   }else{
                      menuPadre.getMenus().add(menu);
                      manager.guardarObjeto(menuPadre);
                      menu.setIdPadre(menuPadre.getId());
                      manager.guardarObjeto(menu);
                      msg.add(ActionMessages.GLOBAL_MESSAGE, new
                   ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION
                 _EXITOSA, menuPadre.getNombre()+": "+menu.getNombre()));
                   DynaActionForm dynaForm = (DynaActionForm) form;
                   dynaForm.set("menu", menuPadre);
                   request.setAttribute("menu", menuPadre);
                   saveMessages(request, msg);
                   return mapping.findForward("editarMenu");
}
           }else{
           msg.add(ActionMessages.GLOBAL_MESSAGE, new
          ActionMessage(Mensajes.APP_MENSAJE_ERROR,
             "El menu ya ha sido asignado a este rol anteriormente."));
            }
     }else{
          msg.add(ActionMessages.GLOBAL_MESSAGE, new
         ActionMessage(Mensajes.APP_MENSAJE_ERROR,
         "No se debe asignar el mismo menu como padre: "+id_menu_padre));
     }
}else{
msg.add(ActionMessages.GLOBAL_MESSAGE, new
ActionMessage(Mensajes.APP_MENSAJE_ERROR, "Escoja un submenu
    para asignarlo al menu."));
}
}else{
   msg.add(ActionMessages.GLOBAL_MESSAGE, new
```

```
   ActionMessage(Mensajes.APP_MENSAJE_ERROR, "Escoja un menu
        para agregar un submenu."));
 }
 saveMessages(request, msg);
  return editarMenu(mapping,form,request,response);
}

public ActionForward quitarSubMenu(ActionMapping mapping, ActionForm
       form, HttpServletRequest request, HttpServletResponse response)
       throws Exception {
       String accion = (String) request.getSession().getAttribute(ACCION);
       if (accion == null || accion.length() <= 0){
          request.getSession().setAttribute(ACCION, SECURITY_ACCION);
       }
       String id_menu_padre = request.getParameter("id_menu_padre");
       ActionMessages msg = new ActionErrors();
       if (id_menu_padre != null && id_menu_padre.length() > 0) {
             String id_menu = request.getParameter("id_menu");
              if (id_menu != null && id_menu.length() > 0) {
                 Menu menuPadre = manager.findMenuById(id_menu_padre);
                 if (menuPadre!= null &&
                   menuPadre.getId().compareTo(Long.valueOf(id_menu)) != 0) {
                       Menu menu = null;
                       for (Iterator iter = menuPadre.getMenus().iterator();
                        iter.hasNext();) {
                            Menu menu2 = (Menu) iter.next();
                            System.out.println(menu2.getNombre());
                            if (menu2.getId().compareTo(Long.valueOf(id_menu)) == 0) {
                                menu = menu2;
                                break;
                            }
                       }
                       if(menu != null && menuPadre.getMenus().contains(menu)){
                          menuPadre.getMenus().remove(menu);
                         manager.guardarObjeto(menuPadre);
                          menu.setIdPadre(null);
                          manager.guardarObjeto(menu);
                          msg.add(ActionMessages.GLOBAL_MESSAGE, new
                     ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION
                           _EXITOSA, menuPadre.getNombre()+":
                           "+menu.getNombre()));
                                DynaActionForm dynaForm = (DynaActionForm) form;
                                dynaForm.set("menu", menuPadre);
                                request.setAttribute("menu", menuPadre);
                                saveMessages(request, msg);
                                return mapping.findForward("editarMenu");
                        }else{
                            msg.add(ActionMessages.GLOBAL_MESSAGE, new
                        ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                        "El menu ya ha sido asignado a este rol anteriormente."));
                       }
                       }else{
                            msg.add(ActionMessages.GLOBAL_MESSAGE, new
```

```
                    ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                    "No se puede quitar el menu que esta editando:
                         "+id_menu_padre));
                    }
                    }else{
                        msg.add(ActionMessages.GLOBAL_MESSAGE, new
                    ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                       "Escoja un submenu para asignarlo al menu."));
                    }
                    }else{
                       msg.add(ActionMessages.GLOBAL_MESSAGE, new
                    ActionMessage(Mensajes.APP_MENSAJE_ERROR,
                      "Escoja un menu para agregar un submenu."));
                    }
                 saveMessages(request, msg);
                    return editarMenu(mapping,form,request,response);
}

public ActionForward guardarMenu(ActionMapping mapping, ActionForm
    form, HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    String accion = (String) request.getSession().getAttribute(ACCION);
    if (accion == null || accion.length() <= 0){
      request.getSession().setAttribute(ACCION, SECURITY_ACCION);
    }
    DynaActionForm dynaForm = (DynaActionForm) form;
    Menu entidad = (Menu) dynaForm.get("menu");
    ActionMessages msg = new ActionErrors();
    String idPadre = request.getParameter("menu.idPadre");
    if (idPadre == null || (idPadre != null && idPadre.length() == 0 ) ){
      entidad.setIdPadre(null);
    }
    if (entidad.getId() != null && entidad.getIdPadre() != null &&
      entidad.getId().compareTo(entidad.getIdPadre()) == 0 ){
      msg.add(ActionMessages.GLOBAL_MESSAGE, new
      ActionMessage(Mensajes.APP_MENSAJE_ERROR,
          "No se debe asignar el mismo menu como padre:
              "+entidad.getIdPadre()));
  }else{
  manager.guardarObjeto(entidad);
  msg.add(ActionMessages.GLOBAL_MESSAGE, new
  ActionMessage(Mensajes.APP_MENSAJE_TRANSACCION
      _EXITOSA, entidad.getNombre()));
}
saveMessages(request, msg);
request.setAttribute("menu", entidad);
return mapping.findForward("editarMenu");
}
}
```

Log4j.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<!DOCTYPE log4j:configuration SYSTEM "./log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern"
        value="%p - %C{1}.%M(%L) | %m%n"/>
    </layout>
  </appender>
  <logger name="net.sf.hibernate">
    <level value="ERROR"/>
  </logger>
  <logger name="org.springframework">
    <level value="ERROR"/>
  </logger>
  <root>
    <level value="ERROR"/>
    <appender-ref ref="CONSOLE"/>
  </root>
</log4j:configuration>
```

messages.properties

Mensajes usados en las etiquetas, errores y en sucesos.
Títulos

```
web.ui.titulo= ACADEMICO
web.ui.usuario.consultar=Consulta de Usuarios
web.ui.usuario.info=Información del Usuario
web.ui.usuario.datos=Datos de Autenticación
web.ui.usuario.roles=Roles asignados
web.ui.rol.consultar=Consulta de Roles
web.ui.rol.info=Información del Rol
web.ui.rol.datos=Datos del Rol
web.ui.rol.menus=Menus asignados
web.ui.menu.consultar=Consulta de Menus
web.ui.menu.info=Información del Menú
web.ui.menu.datos=Datos de acceso y control
web.ui.menu.sub_menus=Sub-menus asignados
web.ui.resultado=Resultado
web.ui.criterios=Criterios de Búsqueda
web.ui.usuario.cambio=Cambio de clave
web.ui.usuario.olvido=Olvido Clave
```

Usuario

```
web.ui.usuario.id=Nombre de usuario
web.ui.usuario.clave=Clave
web.ui.usuario.confirmClave=Confirmación de la clave
web.ui.usuario.nombre=Nombre corto
web.ui.usuario.pregunta=Pregunta de recuperación
web.ui.usuario.respuesta=Respuesta Secreta
```

Rol

```
web.ui.rol.nombre=Nombre del rol
web.ui.rol.descripcion=Descripción del menú
```

Menu
web.ui.menu.nombre=Nombre del menú
web.ui.menu.orden=Orden
web.ui.menu.url=Url de acción o permiso
web.ui.menu.id_padre=Menú padre

Usuario/ Mensajes Generales
web.ui.nombre=Nombre
web.ui.descripcion=Descripción
web.ui.url=Url
web.ui.todos=TODOS
web.ui.usuario=Usuario de Ingreso
web.ui.mostrable=Mostrable
web.ui.mostrable.si=SI
web.ui.mostrable.no=NO
web.ui.eliminable.estado=Estado
web.ui.eliminable.estado.activo=ACTIVO
web.ui.eliminable.estado.inactivo=INACTIVO
web.ui.agrupados=Mostrar menus agrupados
web.ui.agrupados.si=AGRUPADOS
web.ui.agrupados.no=NO AGRUPADOS

Botones
web.ui.boton.crear=Agregar
web.ui.boton.quitar=Quitar
web.ui.boton.consultar=Consultar
web.ui.boton.editar=Editar
web.ui.boton.grabar=Grabar
web.ui.boton.aceptar=Aceptar
web.ui.boton.cancelar=Cancelar
web.ui.boton.eliminar=Eliminar
web.ui.boton.buscar=Buscar
web.ui.boton.busqueda_avanzada=Búsqueda Avanzada
web.ui.boton.calcular=Calcular
web.ui.boton.supervisar=Supervisar

Mensajes Generales
app.mensaje.guardado=El registro de "{0}" ha sido guardado.
app.mensaje.borrado=El registro ha sido borrado. {0}
app.mensaje.no_existe=El registro no existe. {0}
app.mensaje.nombres_no_coinciden=Nombres no coinciden. Nuevo Nombre: {0}, Confirmación: {1}
app.mensaje.transaccion_exitosa=Transacción Exitosa. {0}
app.mensaje.no_userlogin=No se ha autenticado, por favor ingresu su usuario y clave. {0}
app.mensaje.cambio_nombre_no_autorizado=Cambio de nombre no autorizado. {0}
app.mensaje.cambio_nombre_no_nombre_anterior=No se puede autorizar cambio a el mismo nombre de la nave. {0}

Mensajes de Error
app.mensaje.error=La aplicación no puede procesar su requerimiento. {0}
app.mensaje.error.excepcion=La excepción encontrada es: {0}

Validación de errors

errors.required={0} es requerido.
errors.minlength={0} no puede ser menor que {1} caracteres.
errors.maxlength={0} no puede ser mayor que {1} caracteres.
errors.invalid={0} es inválido.
errors.byte={0} debe ser de tipo numérico.
errors.short={0} debe ser de tipo numérico.
errors.integer={0} debe ser un entero.
errors.long={0} debe ser un entero largo.
errors.float={0} debe ser de tipo real.
errors.double={0} debe ser de tipo real.
errors.date={0} no es una fecha.
errors.range={0} no está en el rango desde {1} hasta {2}.
errors.creditcard={0} no es una tarjeta de crédito válida.
errors.email={0} no es una dirección de correo electrónico válida.

404.jsp

```
<%@ page language="java" %>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<html>
<head>
   <%@ include file="/web/jsp/include/include.jsp" %>
</head>
<body>
<table width="100%"  border="0" cellspacing="0" cellpadding="0">
  <tr>
   <td>
   <div id="line_logo">
   <table width="100%" border="0">
          <tr>
                 <td width="38"><img src="web/img/logo.gif"></td>
                 <td width="100%"><span class="texto_left"><bean:message
                 key="web.ui.titulo" /></span></td>
          </tr> </table> </div> </td>
  </tr>
  <tr><td bgcolor="#285577" height="1"/></tr>
  <tr>
   <td align="right"><div id="line_cerrar"><a href="#"
          onclick="window.close()" class="mhTextEmph">Cerrar</a></div></td>
  </tr>
  <tr><td height="10"/></tr>
  <tr>
   <td class="nombre_azul">Informaci&oacute;n no disponible</td>
  </tr>
  <tr><td height="5"/></tr>
  <tr>
   <td>
      <table cellspacing="0" cellpadding="0" border="0" width="100%">
         <tr>
            <td width="2px"/>
            <td width="110px" align="center" valign="middle">
               <img src="web/img/advertencia_azul.png" >
```

```
                </td>
                <td width="8px"/>
                <td class="descripcion_portada" valign="top">
                Lo sentimos mucho. La informaci&oacute;n que usted est&aacute;
                        solicitando no fue encontrada.<br>
                <br>
                La direcci&oacute;n requerida no se encuentra en este servidor.<br>
                <br>
                <br>
                Informaci&oacute;n adicional, por favor, comun&iacute;quese con
                        su administrador de servicios.
                </td>
                <td width="2px"/>
            </tr> </table>  </td>
  </tr>
  <tr><td height="10"/></tr>
</table>
<table align="center">
    <tr>
        <td colspan="3">
            <div id="legal_popup" align="center">
                <span class="copyright">&copy; Sistema de Administraci&oacute;n
                        Acad&eacute;mica</span>
            </div>  </td>  </tr>  </table> </body>
</html>
```

500.jsp

```
<%@ page language="java" isErrorPage="true"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<html>
<head>
    <%@ include file="/web/jsp/include/include.jsp" %>
</head>
<body>
<table width="100%"  border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>
    <div id="line_logo">
    <table width="100%" border="0">
            <tr>
                        <td width="38"><img src="web/img/logo.gif"></td>
                        <td width="100%"><span class="texto_left"><bean:message
                        key="web.ui.titulo" /></span></td>
            </tr>   </table>  </div>  </td>
  </tr>
  <tr><td bgcolor="#285577" height="1"/></tr>
  <tr>
    <td align="right"><div id="line_cerrar"><a href="#"
            onclick="window.close()" class="mhTextEmph">Cerrar</a></div></td>
  </tr>
  <tr><td height="10"/></tr>
  <tr>
```

```
    <td class="nombre_azul">Informaci&oacute;n no disponible</td>
  </tr>
  <tr><td height="5"/></tr>
  <tr>
    <td>
      <table cellspacing="0" cellpadding="0" border="0" width="100%">
        <tr>
          <td width="2px"/>
          <td width="110px" align="center" valign="top">
            <img src="web/img/advertencia_amarilla.png" >
          </td>
          <td width="8px"/>
          <td class="descripcion_portada" valign="top">
          Lo sentimos mucho. Este servicio no est&aacute; disponible en  este momento, por
          favor int&eacute;ntelo m&aacute;s tarde.<br>
          <br>
          Por favor comunique este mensaje a su administrador de servicios:
          <% if (exception != null) { %>
                <blockquote><h6><%= exception.getMessage()
                %></h6></blockquote>
                        <% } else { %>
                                <bean:message key="app.error.contact" />
                        <% } %>
          <br>
          Informaci&oacute;n adicional, por favor, comun&iacute;quese con su administrador
          de servicios.
           <% if (exception != null) { %>
                  <div bgcolor="white" width="90%">
                  <pre><small><% exception.printStackTrace(new
                        java.io.PrintWriter(out)); %></small></pre>
                  </div>
           <% } else { %>
                <bean:message key="app.error.contact" />
           <% } %>
          </td>
          <td width="2px"/>
        </tr>  </table> </td>
  </tr>
  <tr><td height="10"/></tr>
</table>
<table align="center">
  <tr>
    <td colspan="3">
      <div id="legal_popup" align="center">
        <span class="copyright">&copy; Sistema de Administraci&oacute;n
                Acad&eacute;mica</span>
       </div> </td>  </tr> </table> </body>
</html>
```

botones.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%>
   <c:if test="${not empty botones}" >
```

```
<table border="0" >
<tr style="text-align:center">
 <c:forEach items="${botones}" var="boton">
 <td style="text-align:center">
 <c:choose>
    <c:when test="${boton.conFormulario}">
      <c:set var="target" value="iFrameDetalle"/>
      <c:if test="${not empty boton.target}">
       <c:set var="target" value="${boton.target}"/>
      </c:if>
      <html:form action="${boton.accion}" target="${target}">
      <html:hidden property="method" value="${boton.metodo}" />
      <html:submit styleClass="button"
       onclick="this.form.method.value='${boton.metodo}'">${boton.titulo}
             </html:submit>
       <c:forEach items="${boton.parametrosOcultos}" var="parametro">
        <html:hidden property="${parametro.nombre}"
          value="${parametro.valor}" />
       </c:forEach>
     </html:form>
    </c:when>
    <c:otherwise>
      <html:submit styleClass="button"
       onclick="this.form.method.value='${boton.metodo}'">${boton.titulo}
           </html:submit>
       <c:forEach items="${boton.parametrosOcultos}" var="parametro">
        <html:hidden property="${parametro.nombre}"  value="${parametro.valor}" />
       </c:forEach>  </c:otherwise>   </c:choose>  </td>  </c:forEach> </tr>
</table>
</c:if>
```

include.jsp

```
<%
   request.setAttribute("contextPath", request.getContextPath());
   request.setAttribute("datePattern", "dd-MM-yyyy");
%>
 <title>Sistema de Administraci&oacute;n Acad&eacute;mica</title>
                <meta http-equiv="pragma" content="no-cache">
                <meta http-equiv="cache-control" content="no-cache">
                <meta http-equiv="expires" content="0">
<%
response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
response.setHeader("Pragma","no-cache"); //HTTP 1.0
response.setDateHeader ("Expires", 0); //prevents caching at the proxy server
%>
<link type="text/css" rel="stylesheet" href="${contextPath}/web/css/estilo.css"> </link>
<link type="text/css" rel="stylesheet" href="${contextPath}/web/css/screen.css"
media="screen, print" />
<link type="text/css" rel="stylesheet" href="${contextPath}/web/css/footer.css"> </link>
<link type="text/css" rel="stylesheet" href="${contextPath}/web/css/portal.css"> </link>
<link type="text/css" rel="stylesheet" href="${contextPath}/web/css/popup.css"> </link>
```

COMPONENTE PARA AGREGAR CALENDAR POPUP

```
<scri pt language=JavaScriptsrc="${contextPath}/web/js/calendar/popup/
  calendarPop
  up.js"></script>
 <script language=JavaScript
  src="${contextPath}/web/js/calendar/popup/common.js"></script>
  <script language=JavaScript>
  document.write(getCalendarStyles());
</script>
```

mensajes.jsp

Error Messages

```
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<logic:messagesPresent>
<html:messages id="error">
<script type="text/javascript">
    alert('${error}');
</script>
</html:messages>
</logic:messagesPresent>
```

Success Messages

```
<logic:messagesPresent message="true">
   <html:messages id="message" message="true">
   <script type="text/javascript">
       alert('${message}');
   </script>
   </html:messages>
</logic:messagesPresent>
```

pagina_mensajes.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp" %>
<html>
<head>
  <%@ include file="/web/jsp/include/include.jsp" %>
</head>
  <body topmargin="0" >
 <table width="100%" border="0" cellspacing="2" cellpadding="2">
```

Error Messages

```
<logic:messagesPresent>
<tr>
<td class="error">Error:<br/>
<html:messages id="error">
  ${error}<br/>
</html:messages>
</td>
<tr>
</logic:messagesPresent>
```

Success Messages

```
<logic:messagesPresent message="true">
<tr>
<td class="mensaje">Mensaje:<br/>
<html:messages id="message" message="true">
${message}<br/>
</html:messages> </td>
<tr>
</logic:messagesPresent>
</table>  </body>  </html>
```

taglib.jsp

```
<%@ page contentType="text/html;charset=windows-1252"%>
<%@ taglib uri="http://jakarta.apache.org/taglibs/string-1.1" prefix="str" %>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<%@ taglib uri="http://displaytag.sf.net" prefix="display"%>
```

cambio_clave.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
   <head>
      <%@ include file="/web/jsp/include/include.jsp" %>
   </head>
   <body >
      <table width="100%" border="0" cellspacing="2" cellpadding="1"
                  style="margin-top:0 ">
      <c:if test="${empty ACCION}"><c:set var="ACCION"
                  value="security"/></c:if>
      <html:form action="${ACCION}" onsubmit="document.forms[0].name =
                  'usuarioForm';return validateUsuarioForm(this)">
      <html:hidden property="method" value="guardarUsuario" />
      <html:hidden property="ejecutar_metodo" value="${METODO}" />
      <html:hidden property="id_usuario" value="${usuario.id}"/>
      <html:hidden property="isOlvido" value="${param['isOlvido']}"/>
      <html:hidden property="j_u" value="${param['j_u']}"/>
<tr>
      <td class="cab_tabla">
          <table width="100%" border="0">
<tr>
      <td class="cab_tabla_no_borde" width="90%" ><bean:message
              key="web.ui.usuario.info"/></td>
      <td align="right">
      <html:submit styleClass="button"
          onclick="this.form.method.value='guardarUsuario'"><bean:message
            key="web.ui.boton.grabar"/></html:submit>
        </td>
          <td width="1%" />
</tr> </table> </td>  </tr>
        <tr>
```

```
        <td class="cab_tabla">
</td>
            </tr>
           <tr>
               <td class="cab_tabla">
                   <table width="100%" border="0" >
                       <tr>
                           <td class="cab_tabla" colspan="2"><bean:message
                                   key="web.ui.usuario.cambio"/></td>
                       </tr>
                       <tr>
                           <td class="cab_form" width="20%"><bean:message
                                   key="web.ui.usuario.id"/></td>
                           <td class="det_form">${usuario.id}</td>
                       </tr>
                       <tr>
                     <td class="cab_form"><bean:message
                                   key="web.ui.usuario.clave"/></td>
                    <td class="det_form"><html:password
                         property="usuario.clave" size="10" maxlength="16"/></td>
                     </tr>
                       <tr>
                           <td class="cab_form"><bean:message
                         key="web.ui.usuario.confirmClave"/></td>
                           <td class="det_form"><html:password
                         property="usuario.confirmClave" size="10"
                                   maxlength="16"/></td>
                       </tr>
                       <tr>
                           <td class="cab_form"><bean:message
                                   key="web.ui.usuario.nombre"/></td>
                           <td class="det_form">${usuario.nombre}</td>
                       </tr>

                       <html:hidden property="usuario.nombre"
                                   value="${usuario.nombre}"/>
                       <html:hidden property="usuario.pregunta"
                                   value="${usuario.pregunta}"/>
                       <html:hidden property="usuario.respuesta"
                                   value="${usuario.respuesta}"/>
                       <html:hidden property="usuario.estado"
                                   value="${usuario.estado}" />
                   </table>  </td>   </tr>  </html:form>  </table> </body>
</html>
```

Validacion

```
<html:javascript formName="usuarioForm"/>
```

Mensajes

```
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

editar_menu.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
```

```
<head>
       <%@ include file="/web/jsp/include/include.jsp" %>
</head>
<body >
  <table width="100%" border="0" cellspacing="2"
     cellpadding="1"style="margin-top:0 ">
  <c:if test="${empty ACCION}"><c:set var="ACCION"  value="security"/></c:if>
   <html:form action="${ACCION}" onsubmit="document.forms[0].name =
       'menuForm';return validateMenuForm(this)">
      <html:hidden property="method" value="guardarUsuario" />
      <html:hidden property="ejecutar_metodo" value="${METODO}" />
      <html:hidden property="id_menu" value="${menu.id}"/>
<tr>
  <td class="cab_tabla">
  <table width="100%" border="0">
  <tr>
     <td class="cab_tabla_no_borde" width="90%" ><bean:message
          key="web.ui.menu.info"/></td>
         <td align="right">
            <html:submit styleClass="button"
               onclick="this.form.method.value='guardarMenu'"><bean:message
              key="web.ui.boton.grabar"/></html:submit>
         </td>
      <td width="1%" />
  </tr>  </table>  </td>  </tr>
<tr>
<td class="cab_tabla">
</td>  </tr>
<tr>
    <td class="cab_tabla">
      <table width="100%" border="0" >
     <tr>
       <td class="cab_tabla" colspan="2"><bean:message
          key="web.ui.menu.datos"/></td>
     </tr>
<tr>
  <td class="cab_form"><bean:message key="web.ui.menu.nombre"/></td>
  <td class="det_form"><html:text property="menu.nombre"
     size="40" maxlength="100"/></td>
</tr>
<tr>
  <td class="cab_form"><bean:message           key="web.ui.menu.id_padre"/> </td>
  <td class="det_form"><html:text property="menu.idPadre" size="7"
     maxlength="7" readonly="true"/> <INPUT type="text" name="nombreMenu"
     id="nombreMenu" size="20" maxlength="100" readonly="readonly"/>
    <html:button styleClass="button" onclick="var   popup=window.open('${contextPath}
      /security.do?method=listarMenus &  id_menu=${menu.id}&column=menu.idPadre
      &desc=nombreMenu', 'popup','height=400,width=800,status=yes,resizable=yes,
      toolbar=no, menubar=no,scrollbars=yes,location=no');if(popup){popup.focus();};
      "property=""><bean:message key="web.ui.boton.buscar"/>
    </html:button> <html:button styleClass="button"         onclick="if(document
      .all['menu.idPadre']) {document.all['menu.idPadre'].value='';document.all['menu.idPadre']
      .disabled='disabled';};if(document.all['nombreMenu']) {document.all['nombreMenu']
```

```
      .value='';" property=""><bean:message key="web.ui.boton.quitar"/></html:button></td>
 </tr>
<tr>
    <td class="cab_form"><bean:message key="web.ui.menu.url"/></td>
    <td class="det_form"><html:text property="menu.url" size="40"maxlength="100"/></td>
</tr>
<tr>
   <td class="cab_form"><bean:message key="web.ui.menu.orden"/></td>
   <td class="det_form"><html:text property="menu.orden" size="3" maxlength="3"
   /></td>  </tr>
<tr>
   <td class="cab_form" ><bean:message key="web.ui.mostrable"/></td>
   <td class="det_form" >
      <html:select property="menu.mostrable">
      <html:option value="S"><bean:message key="web.ui.mostrable.si"/></html:option>
        <html:option value="N"><bean:message key="web.ui.mostrable.no"/></html:option>
        </html:select> </td>
</tr>
<tr>
 <td class="cab_form"><bean:message key="web.ui.eliminable.estado"/></td>
 <td class="det_form">
   <html:select property="menu.estado">
   <html:option value="A"><bean:message key="web.ui.eliminable.estado.activo"/>
     </html:option>
   <html:option value="I"><bean:message key="web.ui.eliminable.estado.inactivo"/>
      </html:option>
   </html:select>  </td>  </tr> </table>  </td>  </tr>  </html:form>
<c:if test="${!menu.new}">
<c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
<html:form action="${ACCION}" onsubmit="document.forms[0].name =
    'menuForm';return validateMenuForm(this)">
<html:hidden property="method" value="guardarUsuario" />
<html:hidden property="ejecutar_metodo" value="${METODO}" />
<html:hidden property="id_menu_padre" value="${menu.id}"/>
<tr>
    <td class="cab_tabla">
      <table width="100%" border="0">
        <tr>
        <td class="cab_tabla_no_borde" width="90%" ><bean:message
             key="web.ui.menu.sub_menus"/></td>
        <td align="right">
        <html:button styleClass="button" onclick="var popup = window.open('${contextPath}
          /security.do?method=listarMenus&id_menu_padre=${menu.id}','popup','height=400,
           width=800,status=yes,resizable=yes,toolbar=no,menubar=no,scrollbars=yes,
          location=no');
          if(popup){popup.focus();};"property=""><bean:message
              key="web.ui.boton.crear"/></html:button>
        </td>
        <td width="1%" />
      <c:if test="${not empty menu.menus}">
        <td align="right">
          <html:submit styleClass="button" onclick="this.form.method.
             value='quitarSubMenu'"><bean:messagekey="web.ui.boton.quitar"/>
```

```
            </html:submit> </td>
          <td width="1%" />
        </c:if>  </tr>  </table> </td>  </tr>
<tr>
<td class="cab_tabla">
  <display:table name="${menu.menus}"
        id="item"
        class="esmena"
        requestURI="${ACCION}.do?method=editarMenu"
        style="width: 100%"
        sort="list"
        pagesize="100"
        defaultorder="ascending"
        defaultsort="2"
        export="true">
  <display:column title="Escoger" style="text-align: center">
        <input type="radio"
        name="id_menu"
        value="${item.id}"
        checked="checked"></input>
  </display:column>
  <display:column property="id"
        title="C&oacute;digo"
        style="text-align: left; white-space: nowrap"
        maxLength="40"/>
  <display:column property="nombre"
        title="Nombre"
        sortable="true"
        style="text-align: left; white-space: nowrap"
        maxLength="40"/>
  <display:column property="url"
        title="Url"
        style="text-align: left; white-space: nowrap"
        maxLength="40"/>
  <display:column title="Estado"
        style="text-align: center; white-space: nowrap"><c:if test="${item.estado ==
        'A'}"><bean:message key="web.ui.eliminable.estado.activo"/></c:if>
        <c:if test="${item.estado == 'I'}"><bean:message
        key="web.ui.eliminable.estado.inactivo"/></c:if></display:column>
  </display:table>
 </td>
   </tr> </html:form> </c:if> </table>  </body>   </html>
```

Validacion

```
<html:javascript formName="menuForm"/>
<%-- Mensajes --%>
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

editar_rol.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
    <head>
        <%@ include file="/web/jsp/include/include.jsp" %>
    </head>
```

```
<body >
    <table width="100%" border="0" cellspacing="2"
    cellpadding="1" style="margin-top:0 ">
<c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
<html:form action="${ACCION}" onsubmit="document.forms[0].name = 'rolForm';return
   validateRolForm(this)">
    <html:hidden property="method" value="guardarRol" />
    <html:hidden property="ejecutar_metodo" value="${METODO}" />
   <html:hidden property="id_rol" value="${rol.id}"/>
    <tr>
           <td class="cab_tabla">
               <table width="100%" border="0">
                   <tr>
                       <td class="cab_tabla_no_borde" width="90%" ><bean:message
                         key="web.ui.rol.info"/></td>
                       <td align="right">
                       <html:submit styleClass="button"
                           onclick="this.form.method.value='guardarRol'"><bean:message
                           key="web.ui.boton.grabar"/></html:submit>
                       </td>
                       <td width="1%" />
                   </tr> </table> </td>
        </tr>
        <tr>
           <td class="cab_tabla">
      </td>
        </tr>
        <tr>
           <td class="cab_tabla">
               <table width="100%" border="0" >
      <tr>
      <td class="cab_tabla" colspan="2"><bean:message key="web.ui.rol.datos"/></td>
       </tr>
       <tr>
          <td class="cab_form" width="20%"><bean:message
             key="web.ui.rol.nombre"/></td>
          <td class="det_form"><html:text property="rol.nombre" size="40"
             maxlength="100"/></td>
       </tr> <tr>
       <td class="cab_form"><bean:message key="web.ui.rol.descripcion"/></td>
       <td class="det_form"><html:text property="rol.descripcion" size="40"
         maxlength="250"/></td>
        </tr> <tr>
         <td class="cab_form"><bean:message key="web.ui.eliminable.estado"/></td>
         <td class="det_form">
         <html:select property="rol.estado">
         <html:option value="A"><bean:message
              key="web.ui.eliminable.estado.activo"/></html:option>
         <html:option value="I"><bean:message
             key="web.ui.eliminable.estado.inactivo"/></html:option>
         </html:select>  </td> </tr> </table> </td> </tr>
   </html:form>
   <c:if test="${!rol.new}">
```

```
<html:form action="${ACCION}" onsubmit="document.forms[0].name =
      'rolForm';return validateRolForm(this)">
      <html:hidden property="method" value="quitarMenu" />
      <html:hidden property="ejecutar_metodo" value="${METODO}" />
      <html:hidden property="id_rol" value="${rol.id}"/>
   <tr>
   <td class="cab_tabla">
   <table width="100%" border="0">
   <tr>
   <td class="cab_tabla_no_borde" width="90%" ><bean:message
      key="web.ui.rol.menus"/></td>
      <td align="right">
      <html:button styleClass="button" onclick="var popup = window
         .open('${contextPath}/security.do?method=listarMenus&id_rol=${rol.id}',
         'popup','height=400,width=800,status=yes,resizable=yes,toolbar=no,
         menubar=no,scrollbars=yes,location=no');if(popup){popup.focus();};"
         property=""><bean:message key="web.ui.boton.crear"/></html:button>
     </td>
     <td width="1%" />
    <c:if test="${not empty menus}">
   <td align="right">
  <html:submit styleClass="button" onclick="this.form.method.value='quitarMenu'">
<bean:message key="web.ui.boton.quitar"/></html:submit>
</td>
   <td width="1%" />
   </c:if>  </tr>  </table>  </td>  </tr>
   <tr>
     <td class="cab_tabla">
      <display:table name="${menus}"
         id="item"
         class="esmena"
         requestURI="${ACCION}.do?method=editarRol"
         style="width: 100%"
         sort="list"
         pagesize="100"
         defaultorder="ascending"
         defaultsort="2"
         export="true">
     <display:column title="Escoger" style="text-align: center">
         <input type="radio"
         name="id_menu"
         value="${item.id}"
         checked="checked"></input>
     </display:column>
         <display:column property="id"
         title="C&oacute;digo"
         style="text-align: left; white-space: nowrap"
         maxLength="40"/>
     <display:column property="nombre"
         title="Nombre"
         sortable="true"
         style="text-align: left; white-space: nowrap"
         maxLength="40"/>
```

```
<display:column property="url"
      title="Url"
      sortable="true"
      style="text-align: left; white-space: nowrap"
      maxLength="40"/>
<display:column title="Estado"
     style="text-align: center; white-space: nowrap"><c:if test="${item.estado =='A'}">
     <bean:message key="web.ui.eliminable.estado.activo"/>
      </c:if><c:if test="${item.estado == 'I'}"><bean:message
        key="web.ui.eliminable.estado.inactivo"/></c:if>
      </display:column>
<display:column title="Menus"
      style="text-align: left; white-space: nowrap">
      <c:set var="sw" value="${false}"/>
         <c:forEach items="${item.menus}" var="menu1">
            <c:forEach items="${rol.menus}" var="menu2">
               <c:if test="${menu1.id == menu2.id}">
                  <c:set var="sw" value="${true}"/>
               </c:if>  </c:forEach> </c:forEach>
      <c:if test="${sw}">
   <display:table name="${item.menus}"
      id="menu"
      class="esmena"
      style="width: 100%"
      sort="list"
      pagesize="100"
      defaultorder="ascending"
      defaultsort="3"
      export="false"
  >
  <display:setProperty name="paging.banner.all_items_found" value=""/>
  <display:setProperty name="paging.banner.page.link" value=""/>
  <display:setProperty name="basic.msg.empty_list" value=""/>
  <display:setProperty name="paging.banner.full" value=""/>
  <display:setProperty name="paging.banner.first" value=""/>
  <display:setProperty name="paging.banner.last" value=""/>
  <display:setProperty name="paging.banner.page.selected" value=""/>
<c:forEach items="${rol.menus}" var="menu1">
  <c:if test="${menu.id == menu1.id}">
     <display:column property="nombre"
       title="Nombre"
       style="text-align: left; white-space: nowrap"
       maxLength="15"/>
     <display:column property="url"
       title="URL"
       style="text-align: left; white-space: nowrap"
       maxLength="40"/>
     <display:column property="orden"
       title="Orden"
       style="text-align: center; white-space: nowrap"/>
     <display:column title="Quitar"
       style="text-align: center; white-space: nowrap"><input type="checkbox"
         value="${menu.id}" name="id_sub_menu_rol" onclick="location.href=
```

```
'${contextPath}/security.do?method=quitarSubmenuRol&id_rol=
 ${rol.id}&id_sub_menu_rol=${menu.id}'"></display:column>
</c:if>
</c:forEach> </display:table> </c:if> </display:column>
</display:table>
</td> </tr>  </html:form> </c:if> </table>  </body>  </html>
```

Validacion

```
<html:javascript formName="rolForm"/>
<%-- Mensajes --%>
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

editar_usuario.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
  <head>
    <%@ include file="/web/jsp/include/include.jsp" %>
  </head>
  <body >
    <table width="100%" border="0" cellspacing="2"
    cellpadding="1" style="margin-top:0 ">
    <c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
    <html:form action="${ACCION}" onsubmit="document.forms[0].name =
      'usuarioForm';return validateUsuarioForm(this)">
    <html:hidden property="method" value="guardarUsuario" />
    <html:hidden property="ejecutar_metodo" value="${METODO}" />
    <html:hidden property="id_usuario" value="${usuario.id}"/>
    <html:hidden property="clave_anterior" value="${usuario.clave}"/>
     <tr>
         <td class="cab_tabla">
           <table width="100%" border="0">
             <tr>
               <td class="cab_tabla_no_borde" width="90%" ><bean:message
                 key="web.ui.usuario.info"/></td>
                <td align="right">
               <html:submit styleClass="button"
                 onclick="this.form.method.value='guardarUsuario'">
                <bean:message key="web.ui.boton.grabar"/></html:submit>
               </td>
               <td width="1%" />
     </tr>  </table> </td>  </tr>
     <tr>
         <td class="cab_tabla">
       </td>  </tr>
      <tr>
         <td class="cab_tabla">
           <table width="100%" border="0" >
             <tr>
               <td class="cab_tabla" colspan="2"><bean:message
                key="web.ui.usuario.datos"/></td>
             </tr>
             <tr>
             <td class="cab_form" width="20%"><bean:message
               key="web.ui.usuario.id"/></td>
```

```
                    <td class="det_form"><html:text property="usuario.id" size="10"
                      maxlength="16"/></td>
                  </tr>
        <tr>
          <td class="cab_form"><bean:message key="web.ui.usuario.clave"/></td>
          <td class="det_form"><html:password property="usuario.clave" size="10"
            maxlength="16"/></td>
        </tr>
        <tr>
        <td class="cab_form"><bean:message key="web.ui.usuario.confirmClave"/></td>
      <td class="det_form"><html:password property="usuario.confirmClave" size="10"
        maxlength="16"/></td>
      </tr>
    <tr>
  <td class="cab_form"><bean:message key="web.ui.usuario.nombre"/></td>
<td class="det_form"><html:text property="usuario.nombre" size="15"
  maxlength="250"/></td>
</tr>
<tr>
<td class="cab_form"><bean:message key="web.ui.usuario.pregunta"/></td>
  <td class="det_form">
<html:select property="usuario.pregunta">
<html:option value="NOMBRE DE MI PERRO?">NOMBRE DE MI PERRO?</html:option>
<html:option value="NOMBRE DE MI MAMA?">NOMBRE DE MI MAMA?</html:option>
<html:option value="NOMBRE DE MI PAPA?">NOMBRE DE MI PAPA?</html:option>
<html:option value="FECHA DE NACIMIENTO DE MI PAPA?">FECHA DE
  NACIMIENTO DE MI PAPA?</html:option>
<html:option value="FECHA DE NACIMIENTO DE MI MAMA?">FECHA DE
   NACIMIENTO DE MI MAMA?</html:option>
   </html:select>  </td>  </tr>
<tr>
   <td class="cab_form"><bean:message key="web.ui.usuario.respuesta"/></td>
   <td class="det_form"><html:password property="usuario.respuesta" size="20"
     maxlength="250" /></td>
</tr>
<tr>
   <td class="cab_form"><bean:message key="web.ui.eliminable.estado"/></td>
      <td class="det_form">
       <html:select property="usuario.estado">
       <html:option value="A"><bean:message
         key="web.ui.eliminable.estado.activo"/></html:option>
       <c:if test="${SESSION_USERLOGIN.loginName != usuario.id}">
       <html:option value="I"><bean:message
         key="web.ui.eliminable.estado.inactivo"/></html:option>
       </c:if>
       </html:select>
    </td>  </tr>  </table>  </td>
</tr>
</html:form>
   <c:if test="${!usuario.new}">
      <html:form action="${ACCION}" >
      <html:hidden property="method" value="quitarRol" />
      <html:hidden property="id_usuario" value="${usuario.id}"/>
```

```
<tr>
  <td class="cab_tabla">
    <table width="100%" border="0">
     <tr>
       <td class="cab_tabla_no_borde" width="90%" ><bean:message
         key="web.ui.usuario.roles"/></td>
         <td align="right">
            <html:button styleClass="button" onclick="var popup =
              window.open('${contextPath}/security.do?method=listarRoles&id_usuario=
              ${usuario.id}','popup','height=400,width=800,status=yes,resizable=yes,
              toolbar=no,menubar=no,scrollbars=yes,location=no');if(popup)
              {popup.focus();};" property=""><bean:message
              key="web.ui.boton.crear"/></html:button>
          </td>
        <td width="1%" />
      <c:if test="${not empty usuario.roles}">
     <td align="right">
   <html:submit styleClass="button"
     onclick="this.form.method.value='quitarRol'"><bean:message
     key="web.ui.boton.quitar"/></html:submit>
  </td>
  <td width="1%" />
  </c:if>
  </tr>   </table>   </td>  </tr>
<tr>
   <td class="cab_tabla">
     <display:table name="${usuario.roles}"
         id="item"
         class="esmena"
         requestURI="${ACCION}.do?method=editarUsuario"
         style="width: 100%"
         sort="list"
        pagesize="100"
        defaultorder="ascending"
        defaultsort="2"
        export="true">
     <display:column title="Escoger" style="text-align: center">
       <input type="radio"
        name="id_rol"
        value="${item.id}"
       checked="checked"></input>
     </display:column>
       <display:column property="id"
       title="C&oacute;digo"
       style="text-align: left; white-space: nowrap"
       maxLength="40"/>
    <display:column property="nombre"
      title="Nombre"
      sortable="true"
      style="text-align: left; white-space: nowrap"
      maxLength="40"/>
    <display:column property="descripcion"
      title="Descripci&oacute;n"
```

```
      style="text-align: left; white-space: nowrap"
      maxLength="40"/>
    <display:column title="Estado"
     style="text-align: center; white-space: nowrap"><c:if test="${item.estado ==
      'A'}"><bean:messagekey="web.ui.eliminable.estado.activo"/></c:if><c:if
      test="${item.estado == 'I'}"><bean:message
      key="web.ui.eliminable.estado.inactivo"/></c:if></display:column>
    </display:table>
    </td> </tr>  </html:form>
</c:if>  </table>  </body>   </html>
```

Validacion

```
<html:javascript formName="usuarioForm"/>
```

Mensajes

```
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

listar_menus.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
  <head>
      <%@ include file="/web/jsp/include/include.jsp" %>
  </head>
 <body >
 <table width="100%" border="0" cellspacing="2" cellpadding="1" style="margin-top:0 ">
   <c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
   <html:form action="${ACCION}">
   <html:hidden property="method" value="listarMenus" />
   <html:hidden property="ejecutar_metodo" value="${METODO}" />
   <c:if test="${param['column'] != null}">
   <html:hidden property="column" value="${param['column']}" />
   <html:hidden property="desc" value="${param['desc']}" />
   </c:if>
   <tr>
     <td class="cab_tabla">
     <table width="100%" border="0">
     <tr>
     <td class="cab_tabla_no_borde" width="90%" ><bean:message
       key="web.ui.menu.consultar"/></td>
    <td align="right">
    <html:submit styleClass="button"
       onclick="this.form.method.value='listarMenus'"><bean:message
       key="web.ui.boton.consultar"/></html:submit>
    </td>
    <td width="1%" />
    <c:if test="${param['id_rol'] == null && param['column'] == null &&
     param['id_menu_padre'] == null}">
   <td align="right">
   <html:button styleClass="button"
     onclick="location.href='${contextPath}/security.do?method=editarMenu'"
     property=""><bean:message key="web.ui.boton.crear"/></html:button>
   </td>
   <td width="1%" />
   </c:if> </tr> </table> </td> </tr>
  <tr>
```

```
  <td class="cab_tabla">
  <table width="100%" border="0" >
  <tr>
  <td class="cab_tabla" colspan="7" width="100%"><bean:message
    key="web.ui.criterios"/></td>
  </tr>
  <tr>
     <td class="cab_form" >
     <html:radio property="cmbTextToFind" value="TODOS" ><bean:message
       key="web.ui.todos"/></html:radio>
     </td>
     <td class="cab_form" >
     <html:radio property="cmbTextToFind" value="NOMBRE" ><bean:message
       key="web.ui.nombre"/></html:radio>
     </td>
     <td class="cab_form" >
     <html:radio property="cmbTextToFind" value="URL"><bean:message
       key="web.ui.url"/></html:radio>
     </td>
     <td class="det_form" width="20%" >
     <html:text property="textToFind" size="30" maxlength="30"/>
     </td>  </tr>
  <tr>
  <td class="cab_form" ><bean:message key="web.ui.eliminable.estado"/></td>
  <td class="det_form" >
  <html:select property="cmbEstado">
  <html:option value="TODOS"><bean:message key="web.ui.todos"/></html:option>
  <html:option value="A"><bean:message
    key="web.ui.eliminable.estado.activo"/></html:option>
 <html:option value="I"><bean:message
   key="web.ui.eliminable.estado.inactivo"/></html:option>
 </html:select>
 </td>
 <td class="cab_form" ><bean:message key="web.ui.mostrable"/></td>
 <td class="det_form" >
 <html:select property="cmbMostrable">
 <html:option value="S"><bean:message key="web.ui.mostrable.si"/></html:option>
 <html:option value="N"><bean:message key="web.ui.mostrable.no"/></html:option>
</html:select> </td> </tr>
<tr>
  <td class="cab_form" ><bean:message key="web.ui.agrupados"/></td>
  <td class="det_form" >
  <html:select property="cmbAgrupados">
  <html:option value="AGRUPADOS"><bean:message
    key="web.ui.agrupados.si"/></html:option>
  <html:option value="NO_AGRUPADOS"><bean:message
    key="web.ui.agrupados.no"/></html:option>
  </html:select> </td>
  <td class="cab_form" > </td>
  <td class="det_form" > </td>
 </tr> </table> </td> </tr>
 <c:if test="${not empty items}">
 <tr>
```

```
<td class="cab_tabla">
<table width="100%" border="0">
<tr>
<td class="cab_tabla_no_borde" width="90%" ><bean:message
  key="web.ui.resultado"/></td>
<c:if test="${param['id_rol'] == null && param['column'] == null &&
  param['id_menu_padre'] == null}">
<td align="right">
<html:submit styleClass="button"
  onclick="this.form.method.value='editarMenu'"><bean:message
  key="web.ui.boton.editar"/></html:submit>
</td>
<td width="1%" />
</c:if>
<c:if test="${param['id_rol'] != null}">
<td align="right">
<html:button styleClass="button" onclick="var popup =
  window.open('${contextPath}/security.do?method=agregarMenu&id_rol=
  ${param['id_rol']}&id_menu='+id_selected('id_menu'),'iFrameAdmin');if(popup)
  {popup.focus();};" property=""><bean:message key="web.ui.boton.crear"/>
  </html:button>
  <html:hidden property="id_rol" value="${param['id_rol']}" />
  </td>
<td width="1%" />
</c:if>
<c:if test="${param['id_menu_padre'] != null}">
<td align="right">
<html:button styleClass="button" onclick="var popup =
  window.open('${contextPath}/security.do?method=agregarSubMenu&
  id_menu_padre=${param['id_menu_padre']}&id_menu='+id_selected('id_menu'),
  'iFrameAdmin');if(popup){popup.focus();};" property=""><bean:message
  key="web.ui.boton.crear"/></html:button>
<html:hidden property="id_menu_padre" value="${param['id_menu_padre']}" />
</td>
<td width="1%" />
</c:if>  </tr>  </table>  </td>  </tr>
<tr>
<td class="cab_tabla">
<c:if test="${empty METODO}"><c:set var="METODO" value="editarMenu"/></c:if>
<display:table name="items"
   id="item"
   class="esmena"
   requestURI="${ACCION}.do?method=listarMenus"
   style="width: 100%"
   sort="list"
   pagesize="100"
   defaultorder="ascending"
   defaultsort="2"
    export="false">
<display:column title="Escoger" style="text-align: center">
  <input type="radio"
  name="id_menu"
  value="${item.id}"
```

```
  checked="checked"></input>
</display:column>
  <display:column title="C&oacute;digo"
  style="text-align: left; white-space: nowrap"
  sortable="true"
  sortProperty="id">
  <c:if test="${param['column'] != null }">
  <a href="#" onClick="javascript:
  opener.document.forms[0]['<c:out value="${param['column']}"/>'].value='<c:out
     value='${item.id}'/>';
 opener.document.forms[0]['<c:out value="${param['desc']}"/>'].value='<c:out
     value='${item.nombre}'/>';
 window.close();">${item.id}</a>
</c:if>
<c:if test="${param['column'] == null}">${item.id}</c:if>
</display:column>
   <display:column title="Nombre"
   sortable="true"
   style="text-align: left; white-space: nowrap"
   maxLength="40"><c:if test="${param['id_rol'] == null && param['column'] == null &&
    param['id_menu_padre'] == null}"><a href="${contextPath}/${ACCION}.do?
    method=${METODO}&id_menu=${item.id}">${item.nombre}, <br>url = ${item.url}</a>
    </c:if><c:if test="${param['id_rol'] != null || param['column'] != null ||
    param['id_menu_padre'] != null}">${item.nombre}, <br>url =
    ${item.url}</c:if></display:column>
 <display:column title="Estado"
   style="text-align: center; white-space: nowrap"><c:if test="${item.estado ==
   'A'}"><bean:message key="web.ui.eliminable.estado.activo"/></c:if><c:if
   test="${item.estado == 'I'}"><bean:message
   key="web.ui.eliminable.estado.inactivo"/></c:if></display:column>
 <display:column title="Menus"
    style="text-align: left; white-space: nowrap">
 <display:table name="${item.menus}"
    id="menu"
    class="esmena"
    style="width: 100%"
    sort="list"
    pagesize="100"
    defaultorder="ascending"
    defaultsort="3"
    export="false"
 >
 <display:setProperty name="paging.banner.all_items_found" value=""/>
 <display:setProperty name="paging.banner.page.link" value=""/>
 <display:setProperty name="basic.msg.empty_list" value=""/>
 <display:setProperty name="paging.banner.full" value=""/>
 <display:setProperty name="paging.banner.first" value=""/>
 <display:setProperty name="paging.banner.last" value=""/>
 <display:setProperty name="paging.banner.page.selected" value=""/>
 <display:column property="nombre"
    title="Nombre"
    style="text-align: left; white-space: nowrap"
    maxLength="15"/>
```

```
  <display:column property="url"
    title="URL"
    style="text-align: left; white-space: nowrap"
    maxLength="40"/>
  <display:column property="orden"
    title="Orden"
    style="text-align: center; white-space: nowrap"/>
  </display:table>
  </display:column>
  </display:table>
  </td>  </tr> </c:if> </html:form> </table> </body> </html>
```

Mensajes

```
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

listar_roles.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
<head>
  <%@ include file="/web/jsp/include/include.jsp" %>
</head>
<body >
  <table width="100%" border="0" cellspacing="2" cellpadding="1" style="margin-
    top:0 "
  <c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
  <html:form action="${ACCION}">
  <html:hidden property="method" value="listarUsuarios" />
  <html:hidden property="ejecutar_metodo" value="${METODO}" />
  <c:if test="${param['id_usuario'] != null}">
  <html:hidden property="id_usuario" value="${param['id_usuario']}" />
  </c:if>
<tr>
  <td class="cab_tabla">
   <table width="100%" border="0">
   <tr>
     <td class="cab_tabla_no_borde" width="90%" ><bean:message
       key="web.ui.rol.consultar"/></td>
     <td align="right">
     <html:submit styleClass="button"
       onclick="this.form.method.value='listarRoles'"><bean:message
       key="web.ui.boton.consultar"/></html:submit>
  </td>
  <td width="1%" />
  <c:if test="${param['id_usuario'] == null}">
  <td align="right">
  <html:button styleClass="button"
   onclick="location.href='${contextPath}/security.do?method=editarRol'"
   property=""><bean:message key="web.ui.boton.crear"/></html:button>
  </td>
  <td width="1%" />
  </c:if>  </tr>  </table> </td></tr>
  <tr>
    <td class="cab_tabla">
     <table width="100%" border="0" >
```

```
<tr>
  <td class="cab_tabla" colspan="7" width="100%"><bean:message
    key="web.ui.criterios"/></td>
</tr>
<tr>
   <td class="cab_form" >
      <html:radio property="cmbTextToFind" value="TODOS" ><bean:message
       key="web.ui.todos"/></html:radio>
   </td>
   <td class="cab_form" >
      <html:radio property="cmbTextToFind" value="NOMBRE" ><bean:message
      key="web.ui.nombre"/></html:radio>
   </td>
   <td class="cab_form" >
    <html:radio property="cmbTextToFind" value="DESCRIPCION"><bean:message
        key="web.ui.descripcion"/></html:radio>
   </td>
   <td class="det_form" width="20%" >
   <html:text property="textToFind" size="30" maxlength="30"/>
</td> </tr>
<tr>
  <td class="cab_form" ><bean:message key="web.ui.eliminable.estado"/></td>
  <td class="det_form" >
  <html:select property="cmbEstado">
  <html:option value="TODOS"><bean:message key="web.ui.todos"/></html:option>
  <html:option value="A"><bean:message        key="web.ui.eliminable.estado.activo"/>
    </html:option>
  <html:option value="I"><bean:message key="web.ui.eliminable.estado.inactivo"/>
    </html:option>
  </html:select>  </td>
  <td class="cab_form" >
  </td>
    <td class="det_form" width="20%" >
  </td> </tr> </table> </td>  </tr>
<c:if test="${not empty items}">
<tr>
  <td class="cab_tabla">
  <table width="100%" border="0">
  <tr>
  <td class="cab_tabla_no_borde" width="90%" ><bean:message
    key="web.ui.resultado"/></td>
  <c:if test="${param['id_usuario'] == null}">
  <td align="right">
    <html:submit styleClass="button"
      onclick="this.form.method.value='editarRol'"><bean:message
      key="web.ui.boton.editar"/></html:submit>
 </td>
 <td width="1%" />
 </c:if>
 <c:if test="${param['id_usuario'] != null}">
 <td align="right">
 <%--
 Aqui tienen q poner en alguan parte algo como window.close();
```

```
--%>
<html:button styleClass="button" onclick="var popup = window.open('${contextPath}
/security.do?method=agregarRol&id_usuario=${param['id_usuario']}&id_rol=
'+id_selected('id_rol'),'iFrameAdmin');if(popup){popup.focus();};" property="">
<bean:message key="web.ui.boton.crear"/></html:button>
</td>
<td width="1%" />
</c:if> </tr> </table>  </td>  </tr>
<tr>
  <td class="cab_tabla">
  <c:if test="${empty METODO}"><c:set var="METODO" value="editarRol"/></c:if>
  <display:table name="items"
    id="item"
    class="esmena"
    requestURI="${ACCION}.do?method=listarRoles"
    style="width: 100%"
    sort="list"
    pagesize="100"
    defaultorder="ascending"
    defaultsort="2"
    export="excel">
  <display:column title="Escoger" style="text-align: center">
    <input type="radio"
    name="id_rol"
    value="${item.id}"
    checked="checked"></input>
  </display:column>
    <display:column property="id"
      title="C&oacute;digo"
      style="text-align: left; white-space: nowrap"
      maxLength="40"/>
    <display:column title="Nombre"
      sortable="true"
      style="text-align: left; white-space: nowrap"
      maxLength="40"><c:if test="${param['id_usuario'] == null}"><a href=
      "${contextPath} /${ACCION}.do?method=${METODO}&id_rol=${item.id}">
      ${item.nombre}</a></c:if><c:if test="${param['id_usuario'] != null}">
      ${item.nombre}</c:if></display:column>
   <display:column property="descripcion"
     title="Descripci&oacute;n"
     style="text-align: left; white-space: nowrap"
     maxLength="40"/>
   <display:column title="Estado"
     style="text-align: center; white-space: nowrap"><c:if test="${item.estado == 'A'}">
     <bean:message key="web.ui.eliminable.estado.activo"/></c:if><c:if
     test="${item.estado == 'I'}"><bean:message
     key="web.ui.eliminable.estado.inactivo"/></c:if></display:column>
 </display:table>
 </td>  </tr> </c:if>  </html:form>   </table>   </body>   </html>
```

Mensajes

```
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

listar_usuarios.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
<head>
  <%@ include file="/web/jsp/include/include.jsp" %>
</head>
<body >
 <table width="100%" border="0" cellspacing="2" cellpadding="1" style="margin-top:0 ">
   <c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
   <html:form action="${ACCION}">
   <html:hidden property="method" value="listarUsuarios" />
   <html:hidden property="ejecutar_metodo" value="${METODO}" />
   <tr>
   <td class="cab_tabla">
   <table width="100%" border="0">
   <tr>
   <td class="cab_tabla_no_borde" width="90%" ><bean:message
     key="web.ui.usuario.consultar"/></td>
     <td align="right">
     <html:submit styleClass="button"
       onclick="this.form.method.value='listarUsuarios'"><bean:message
       key="web.ui.boton.consultar"/></html:submit>
   </td>
   <td width="1%" />
   <td align="right">
   <html:button styleClass="button"
     onclick="location.href='${contextPath}/security.do?method=editarUsuario'"
     property=""><bean:message key="web.ui.boton.crear"/></html:button>
  </td>
  <td width="1%" />
  </tr> </table>  </td>  </tr>
<tr>
  <td class="cab_tabla">
  <table width="100%" border="0" >
  <tr>
    <td class="cab_tabla" colspan="7" width="100%"><bean:message
      key="web.ui.criterios"/></td>
  </tr>
  <tr>
    <td class="cab_form" >
     <html:radio property="cmbTextToFind" value="TODOS" ><bean:message
       key="web.ui.todos"/></html:radio>
    </td>
    <td class="cab_form" >
     <html:radio property="cmbTextToFind" value="NOMBRE" ><bean:message
      key="web.ui.nombre"/></html:radio>
   </td>
   <td class="cab_form" >
     <html:radio property="cmbTextToFind" value="USUARIO"><bean:message
       key="web.ui.usuario"/></html:radio>
   </td>
   <td class="det_form" width="20%" >
     <html:text property="textToFind" size="30" maxlength="30"/>
```

```
  </td>  </tr>
 <tr>
 <td class="cab_form" ><bean:message key="web.ui.eliminable.estado"/></td>
 <td class="det_form" >
   <html:select property="cmbEstado">
   <html:option value="TODOS"><bean:message key="web.ui.todos"/></html:option>
   <html:option value="A"><bean:message
     key="web.ui.eliminable.estado.activo"/></html:option>
   <html:option value="I"><bean:message
     key="web.ui.eliminable.estado.inactivo"/></html:option>
   </html:select>
 </td>
 <td class="cab_form" colspan="2">
 </td> </tr> </table>  </td>  </tr>
<c:if test="${not empty items}">
<tr>
<td class="cab_tabla">
 <table width="100%" border="0">
 <tr>
   <td class="cab_tabla_no_borde" width="90%" ><bean:message
     key="web.ui.resultado"/></td>
   <td align="right">
     <html:submit styleClass="button"
     onclick="this.form.method.value='editarUsuario'"><bean:message
     key="web.ui.boton.editar"/></html:submit>
   </td>
   <td width="1%" />
   </tr>  </table>  </td>  </tr>
   <tr>
     <td class="cab_tabla">
     <c:if test="${empty METODO}"><c:set var="METODO"
     value="editarUsuario"/></c:if>
     <display:table name="items"
       id="item"
       class="esmena"
       requestURI="${ACCION}.do?method=listarUsuarios"
       style="width: 100%"
       sort="list"
       pagesize="100"
       defaultorder="ascending"
       defaultsort="2"
       export="false">
     <display:column title="Escoger" style="text-align: center">
       <input type="radio"
       name="id_usuario"
       value="${item.id}"
        checked="checked"></input>
     </display:column>
       <display:column title="Nombre"
          sortable="true"
         style="text-align: left; white-space: nowrap"
         maxLength="40"><a
         href="${contextPath}/${ACCION}.do?method=${METODO}&id_usuario=
```

```
        ${item.id}">${item.nombre}</a></display:column>
      <display:column property="id"
        title="Usuario"
        style="text-align: left; white-space: nowrap"
        maxLength="40"/>
      <display:column title="Fecha de Creaci&oacute;n"
        style="text-align: center; white-space: nowrap"
        maxLength="40"><fmt:formatDate value="${item.fechaSolicitud}"
        pattern="${datePattern}"/></display:column>
       <display:column title="Estado"
         style="text-align: center; white-space: nowrap"><c:if test="${item.estado ==
          'A'}"><bean:message key="web.ui.eliminable.estado.activo"/></c:if><c:if
          test="${item.estado == 'I'}"><bean:message
         key="web.ui.eliminable.estado.inactivo"/></c:if></display:column>
       </display:table> </td> </tr> </c:if>  </html:form> </table> </body> </html>
```

olvido_clave.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%><html>
<head>
  <%@ include file="/web/jsp/include/include.jsp" %>
</head>
<body >
 <table width="100%" border="0" cellspacing="2" cellpadding="1" style="margin-top:0 ">
  <c:if test="${empty ACCION}"><c:set var="ACCION" value="security"/></c:if>
  <html:form action="${ACCION}" onsubmit="document.forms[0].name =
    'usuarioForm';return validateUsuarioForm(this)">
  <html:hidden property="method" value="olvidoClave" />
  <html:hidden property="id_usuario" value="${usuario.id}"/>
  <html:hidden property="isOlvido" value="${param['isOlvido']}"/>
  <html:hidden property="j_u" value="${param['j_u']}"/>
  <tr>
    <td class="cab_tabla">
      <table width="100%" border="0">
      <tr>
        <td class="cab_tabla_no_borde" width="90%" ><bean:message
           key="web.ui.usuario.info"/></td>
        <td align="right">
          <html:submit styleClass="button"
            onclick="this.form.method.value='olvidoClave'"><bean:message
            key="web.ui.boton.grabar"/></html:submit>
          </td>
        <td width="1%" />
    </tr> </table>  </td>   </tr>
  <tr>
    <td class="cab_tabla">
  </td>  </tr>
  <tr>
    <td class="cab_tabla">
      <table width="100%" border="0" >
        <tr>
          <td class="cab_tabla" colspan="2"><bean:message
            key="web.ui.usuario.olvido"/></td>
```

```
          </tr>
          <tr>
            <td class="cab_form" width="20%"><bean:message
              key="web.ui.usuario.id"/></td>
            <td class="det_form">${usuario.id}</td>
          </tr>
  <tr>
    <td class="cab_form"><bean:message key="web.ui.usuario.nombre"/></td>
    <td class="det_form">${usuario.nombre}</td>
  </tr>
  <tr>
    <td class="cab_form"><bean:message key="web.ui.usuario.pregunta"/></td>
    <td class="det_form">${usuario.pregunta}</td>
  </tr>
  <tr>
    <td class="cab_form"><bean:message key="web.ui.usuario.respuesta"/></td>
    <td class="det_form"><input type="text" name="usuario.respuesta" size="20"
      maxlength="250" /></td>
 </tr>
 <html:hidden property="usuario.nombre" value="${usuario.nombre}"/>
 <html:hidden property="usuario.clave" value="${usuario.clave}"/>
 <html:hidden property="usuario.confirmClave" value="${usuario.confirmClave}"/>
 <html:hidden property="usuario.respuesta" value="${usuario.respuesta}"/>
 <html:hidden property="usuario.estado" value="${usuario.estado}" />
 </table> </td> </tr>  </html:form>  </table>   </body>   </html>
```

Validacion

```
<html:javascript formName="usuarioForm"/>
```

Mensajes

```
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

action-servlet.xml

```
<beans>
        <bean name="/global" class="com.bvg.comun.web.GlobalAction" singleton="false">
        <property name="modelManager"><ref bean="modelManager"/></property>
    </bean>
    <bean name="/security" class="com.bvg.security.web.SecurityAction"
      singleton="false">
   <property name="modelManager"><ref bean="modelManager"/></property>
    </bean>
</beans>
```

aplicationContext.xml

```
<beans>
```

DataSource

```
<bean id="dataSource"
<property
   name="driverClassName"><value>com.microsoft.jdbc.sqlserver.SQLServerDriver
   </value></property>
<property name="url"><value>jdbc:Microsoft:sqlserver://127.0.0.1:1433;
```

```
  DatabaseName=portalweb;SelectMethod=cursor</value></property>
<property name="username"><value>sa</value></property>
<property name="password"><value>zaqxsw80</value></property>
<property name="initialPoolSize" value="5"/>
<property name="minPoolSize" value="5"/>
<property name="maxPoolSize" value="15"/>
<property name="maxIdleTime" value="30"/>
<property name="idleConnectionTestPeriod" value="30"/>
<property name="driverClass" value="org.postgresql.Driver"/>
<property name="jdbcUrl" value="jdbc:postgresql://127.0.0.1:5432/academico"/>
<property name="driverClass" value="com.microsoft.jdbc.sqlserver.SQLServerDriver"/>
<property name="jdbcUrl" value="jdbc:Microsoft:sqlserver://127.0.0.1:1433;
  DatabaseName=academico;SelectMethod=cursor"/>
<property name="user" value="sa"/>
<property name="password" value="tequila"/>
<property name="driverClass" value="com.microsoft.jdbc.sqlserver.SQLServerDriver"/>
<property name="jdbcUrl" value="jdbc:Microsoft:sqlserver://127.0.0.1:1433;
  DatabaseName=academico;SelectMethod=cursor"/>
<property name="driverClassName" value="org.postgresql.Driver"/>
<property name="url" value="jdbc:postgresql://127.0.0.1:5432/postgres"/>
<property name="username" value="postgres"/>
<property name="password" value="postgres"/>
</bean>
```

Hibernate SessionFactory

```
<bean id="sessionFactory"
 class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
<property name="dataSource"><ref local="dataSource"/></property>
<property name="mappingResources">
<list>
```

Se maneja la Seguridad

```
<value>com/bvg/comun/security/model/auth/Log.hbm.xml</value>
<value>com/bvg/comun/security/model/auth/Usuario.hbm.xml</value>
<value>com/bvg/comun/security/model/auth/Rol.hbm.xml</value>
<value>com/bvg/comun/security/model/auth/Menu.hbm.xml</value>
</list>  </property>
<property name="hibernateProperties">
<props>
  <prop key="hibernate.show_sql">false</prop>
  <prop key="hibernate.generate_statistics">false</prop>
  <prop key="hibernate.use_sql_comments">false</prop>
  <prop key="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</prop>
  <prop key="hibernate.statement_cache.size">0</prop>
  <prop key="hibernate.cache.use_query_cache">false</prop>
  <prop key="hibernate.cache.provider_class">org.hibernate.cache
    .HashtableCacheProvider</prop>
  <prop key="hibernate.query.factory_class">org.hibernate.hql.classic
    .ClassicQueryTranslatorFactory</prop>
  </props> </property>
<property name="eventListeners">
<map> <entry key="merge">
<bean class="org.springframework.orm.hibernate3.support.IdTransferringMerge
```

```
  EventListener"/>
</entry></map> </property></bean>

<bean id="transactionManager" class="org.springframework.orm.hibernate3
   .HibernateTransactionManager">
<property name="sessionFactory"><ref local="sessionFactory"/></property>
</bean>

<bean id="hibernateInterceptor" class="org.springframework.orm.hibernate3
   .HibernateInterceptor">
<property name="sessionFactory">
<ref bean="sessionFactory"/>
</property></bean>

<bean id="comunDAO" class="com.bvg.comun.model.dao.ComunDAOHibernate">
   <property name="sessionFactory"><ref local="sessionFactory"/></property>
</bean>

<bean id="managerTarget" class="com.bvg.security.service.Manager">
   <property name="comunDAO"><ref local="comunDAO"/></property>
</bean>

<bean id="businessObject" class="org.springframework.aop.framework
   .ProxyFactoryBean">
<property name="target"><ref bean="managerTarget"/></property>
<property name="proxyInterfaces">
<value>com.bvg.security.service.ModelManager</value>
</property>
<property name="interceptorNames">
<list>
<value>hibernateInterceptor</value>
</list> </property> </bean>

<bean id="modelManager"
   class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
<property name="transactionManager"><ref local="transactionManager"/></property>
<property name="target"><ref local="managerTarget"/></property>
<property name="transactionAttributes">
<props>
<prop key="find*">PROPAGATION_REQUIRED,readOnly</prop>
<prop key="get*">PROPAGATION_REQUIRED,readOnly</prop>
<prop key="guardar*">PROPAGATION_REQUIRED</prop>
<prop key="procesar*">PROPAGATION_REQUIRED</prop>
</props> </property></bean>

<bean id="securityTarget" class="com.bvg.comun.security.dao.SecurityImpl">
   <property name="sessionFactory" ref="sessionFactory"/>
</bean>

<bean id="security" class="org.springframework.transaction.interceptor
   .TransactionProxyFactoryBean">
   <property name="transactionManager" ref="transactionManager"/>
   <property name="target" ref="securityTarget"/>
```

```
  <property name="transactionAttributes">
  <props> <prop key="get*"  >PROPAGATION_REQUIRED,readOnly</prop>
  <prop key="find*" >PROPAGATION_REQUIRED,readOnly</prop>
  <prop key="load*" >PROPAGATION_REQUIRED,readOnly</prop>
  <prop key="store*">PROPAGATION_REQUIRED</prop>
  </props>  </property>  </bean>
</beans>
```

stuts-config.xml

```
<struts-config>
 <form-beans>
  <form-bean
    name="nullForm" type="org.apache.struts.validator.DynaValidatorForm">
  </form-bean>
  <form-bean
   name="securityForm" type="org.apache.struts.validator.DynaValidatorForm">
  <form-property name="usuario" type="com.bvg.comun.security.model.auth.Usuario"/>
  <form-property name="rol" type="com.bvg.comun.security.model.auth.Rol"/>
  <form-property name="menu"type="com.bvg.comun.security.model.auth.Menu"/>
  <form-property name="persona" type="com.bvg.comun.security.model.auth.Persona"/>
  <form-property name="textToFind" type="java.lang.String"/>
  <form-property name="cmbTextToFind" type="java.lang.String"/>
  <form-property name="cmbEstado" type="java.lang.String"/>
  <form-property name="cmbMostrable" type="java.lang.String"/>
  <form-property name="cmbAgrupados" type="java.lang.String"/>
  </form-bean>

  <form-bean
    name="usuarioForm" type="org.apache.struts.validator.DynaValidatorForm">
    <form-property name="usuario" type="com.bvg.comun.security.model.auth.Usuario"/>
  </form-bean>

  <form-bean
    name="rolForm" type="org.apache.struts.validator.DynaValidatorForm">
    <form-property name="rol" type="com.bvg.comun.security.model.auth.Rol"/>
  </form-bean>

  <form-bean
    name="menuForm" type="org.apache.struts.validator.DynaValidatorForm">
    <form-property name="menu" type="com.bvg.comun.security.model.auth.Menu"/>
  </form-bean>

  <form-bean
    name="personaForm" type="org.apache.struts.validator.DynaValidatorForm">
  <form-property name="persona" type="com.bvg.comun.security.model.auth.Persona"/>
  </form-bean>
</form-beans>
```

Global Forward Definitions

```
<global-forwards>
```

Generales

```
<forward name="paginaEnBlanco" path="/web/jsp/inlcude/blanco.jsp"/>
<forward name="paginaMensajes"
  path="/web/jsp/include/pagina_mensajes.jsp"/>
<forward name="login" path="/login.jsp" />
<forward name="logoff" path="/logoff.jsp" />
</global-forwards>
```

Action Mapping Definitions

```
<action-mappings>
 <action path="/global" type="org.springframework.web.struts.DelegatingActionProxy"
  name="nullForm" scope="session" parameter="method" validate="false">
  <forward name="index" path="/intro.jsp" />
</action>

<action path="/security" type="org.springframework.web.struts.DelegatingActionProxy"
  name="securityForm" scope="session" parameter="method" valida   e="false">
```

Persona

```
 <forward name="listarPersonas" path="/web/jsp/security/listar_personas.jsp" />
 <forward name="editarPersona" path="/web/jsp/security/editar_persona.jsp" />
```

Usuarios

```
 <forward name="listarUsuarios" path="/web/jsp/security/listar_usuarios.jsp" />
 <forward name="editarUsuario" path="/web/jsp/security/editar_usuario.jsp" />
 <forward name="buscarEmpleado" path="/web/jsp/security/buscar_empleado.jsp" />
 <forward name="buscarRol" path="/web/jsp/security/buscar_rol.jsp" />
```

Roles

```
 <forward name="listarRoles" path="/web/jsp/security/listar_roles.jsp" />
 <forward name="editarRol" path="/web/jsp/security/editar_rol.jsp" />
 <forward name="buscarMenu" path="/web/jsp/security/buscar_menu.jsp" />
```

Menus

```
 <forward name="listarMenus" path="/web/jsp/security/listar_menus.jsp" />
 <forward name="editarMenu" path="/web/jsp/security/editar_menu.jsp" />
 <forward name="cambioClave" path="/web/jsp/security/cambio_clave.jsp" />
 <forward name="olvidoClave" path="/web/jsp/security/olvido_clave.jsp" />
</action>
</action-mappings>
```

Message Resources Definitions

```
<message-resources parameter="messages"/>
```

Plug Ins Configuration

```
<plug-in
 className="org.springframework.web.struts.ContextLoaderPlugIn">
 <set-property property="contextConfigLocation"
 value="/WEB-INF/applicationContext.xml, /WEB-INF/action-servlet.xml"/>
</plug-in>

<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
  <set-property property="pathnames"
  value="/WEB-INF/validator-rules.xml,  /WEB-INF/validation.xml"/>
```

```
</plug-in>
</struts-config>
```

validation.xml

```
<form-validation>
  <formset>
    <form name="usuarioForm">
      <field property="usuario.id" depends="required">
        <arg position="0" key="web.ui.usuario.id"/>
      </field>
      <field property="usuario.clave" depends="required">
        <arg position="0" key="web.ui.usuario.clave"/>
      </field>
      <field property="usuario.confirmClave" depends="required">
        <arg position="0" key="web.ui.usuario.confirmClave"/>
      </field>
      <field property="usuario.pregunta" depends="required">
        <arg position="0" key="web.ui.usuario.pregunta"/>
      </field>
      <field property="usuario.nombre" depends="required">
        <arg position="0" key="web.ui.usuario.nombre"/>
      </field>
      <field property="usuario.estado" depends="required">
        <arg position="0" key="web.ui.eliminable.estado"/>
      </field>
    </form>

    <form name="rolForm">
      <field property="rol.nombre" depends="required">
        <arg position="0" key="web.ui.rol.nombre"/>
      </field>
      <field property="rol.descripcion" depends="required">
        <arg position="0" key="web.ui.rol.descripcion"/>
      </field>
      <field property="rol.estado" depends="required">
        <arg position="0" key="web.ui.eliminable.estado"/>
      </field>
    </form>

    <form name="menuForm">
      <field property="menu.nombre" depends="required">
        <arg position="0" key="web.ui.menu.nombre"/>
      </field>
      <field property="menu.url" depends="required">
        <arg position="0" key="web.ui.menu.url"/>
      </field>
      <field property="menu.orden" depends="required,integer">
        <arg position="0" key="web.ui.menu.orden"/>
      </field>
      <field property="menu.mostrable" depends="required">
        <arg position="0" key="web.ui.mostrable"/>
      </field>
```

```
    <field property="menu.estado" depends="required">
      <arg position="0" key="web.ui.eliminable.estado"/>
    </field>
    <field property="menu.idPadre" depends="float">
      <arg position="0" key="web.ui.menu.id_padre"/>
    </field>
  </form>
 </formset>
</form-validation>
```

validator-rules.xml

Aquí se muestra las validaciones por default de los Struts, estos son ubicados en algun lugar en el /WEB-INF y referenciados en el struts-config.xml bajo los elementos del plug-in para la validación.

```
<form-validation>
  <global>
    <validator name="required"
          classname="org.apache.struts.validator.FieldChecks"
             method="validateRequired"
      methodParams="java.lang.Object,
                 org.apache.commons.validator.ValidatorAction,
                 org.apache.commons.validator.Field,
                 org.apache.struts.action.ActionMessages,
                 org.apache.commons.validator.Validator,
                 javax.servlet.http.HttpServletRequest"
              msg="errors.required"/>

    <validator name="requiredif"
          classname="org.apache.struts.validator.FieldChecks"
          method="validateRequiredIf"
          methodParams="java.lang.Object,
                   org.apache.commons.validator.ValidatorAction,
                   org.apache.commons.validator.Field,
                   org.apache.struts.action.ActionMessages,
                   org.apache.commons.validator.Validator,
                   javax.servlet.http.HttpServletRequest"
          msg="errors.required"/>

    <validator name="validwhen"
       msg="errors.required"
          classname="org.apache.struts.validator.validwhen.ValidWhen"
          method="validateValidWhen"
          methodParams="java.lang.Object,
             org.apache.commons.validator.ValidatorAction,
             org.apache.commons.validator.Field,
             org.apache.struts.action.ActionMessages,
             org.apache.commons.validator.Validator,
             javax.servlet.http.HttpServletRequest"/>

    <validator name="minlength"
       classname="org.apache.struts.validator.FieldChecks"
```

```
        method="validateMinLength"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends=""
           msg="errors.minlength"
    jsFunction="org.apache.commons.validator.javascript.validateMinLength"/>

<validator name="maxlength"
      classname="org.apache.struts.validator.FieldChecks"
        method="validateMaxLength"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends=""
           msg="errors.maxlength"
    jsFunction="org.apache.commons.validator.javascript.validateMaxLength"/>

<validator name="mask"
      classname="org.apache.struts.validator.FieldChecks"
        method="validateMask"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends=""
           msg="errors.invalid"/>

<validator name="byte"
      classname="org.apache.struts.validator.FieldChecks"
        method="validateByte"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends=""
           msg="errors.byte"
 jsFunctionName="ByteValidations"/>

<validator name="short"
      classname="org.apache.struts.validator.FieldChecks"
        method="validateShort"
  methodParams="java.lang.Object,
```

```
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
        depends=""
            msg="errors.short"
 jsFunctionName="ShortValidations"/>

<validator name="integer"
      classname="org.apache.struts.validator.FieldChecks"
         method="validateInteger"
   methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
        depends=""
            msg="errors.integer"
 jsFunctionName="IntegerValidations"/>

<validator name="long"
      classname="org.apache.struts.validator.FieldChecks"
         method="validateLong"
   methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
        depends=""
            msg="errors.long"/>

<validator name="float"
      classname="org.apache.struts.validator.FieldChecks"
         method="validateFloat"
   methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
        depends=""
            msg="errors.float"
 jsFunctionName="FloatValidations"/>

<validator name="double"
      classname="org.apache.struts.validator.FieldChecks"
         method="validateDouble"
   methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
```

```
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends=""
          msg="errors.double"/>

<validator name="date"
     classname="org.apache.struts.validator.FieldChecks"
        method="validateDate"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends=""
          msg="errors.date"
 jsFunctionName="DateValidations"/>

<validator name="intRange"
     classname="org.apache.struts.validator.FieldChecks"
        method="validateIntRange"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends="integer"
          msg="errors.range"/>

<validator name="floatRange"
     classname="org.apache.struts.validator.FieldChecks"
        method="validateFloatRange"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends="float"
          msg="errors.range"/>

<validator name="doubleRange"
     classname="org.apache.struts.validator.FieldChecks"
        method="validateDoubleRange"
  methodParams="java.lang.Object,
              org.apache.commons.validator.ValidatorAction,
              org.apache.commons.validator.Field,
              org.apache.struts.action.ActionMessages,
              org.apache.commons.validator.Validator,
              javax.servlet.http.HttpServletRequest"
       depends="double"
```

```
            msg="errors.range"/>

     <validator name="creditCard"
          classname="org.apache.struts.validator.FieldChecks"
             method="validateCreditCard"
       methodParams="java.lang.Object,
                     org.apache.commons.validator.ValidatorAction,
                     org.apache.commons.validator.Field,
                     org.apache.struts.action.ActionMessages,
                     org.apache.commons.validator.Validator,
                     javax.servlet.http.HttpServletRequest"
            depends=""
                msg="errors.creditcard"/>

     <validator name="email"
          classname="org.apache.struts.validator.FieldChecks"
             method="validateEmail"
       methodParams="java.lang.Object,
                     org.apache.commons.validator.ValidatorAction,
                     org.apache.commons.validator.Field,
                     org.apache.struts.action.ActionMessages,
                     org.apache.commons.validator.Validator,
                     javax.servlet.http.HttpServletRequest"
            depends=""
                msg="errors.email"/>

     <validator name="url"
          classname="org.apache.struts.validator.FieldChecks"
             method="validateUrl"
       methodParams="java.lang.Object,
                     org.apache.commons.validator.ValidatorAction,
                     org.apache.commons.validator.Field,
                     org.apache.struts.action.ActionMessages,
                     org.apache.commons.validator.Validator,
                     javax.servlet.http.HttpServletRequest"
            depends=""
                msg="errors.url"/>

    <validator name="includeJavaScriptUtilities"
          classname=""
             method=""
       methodParams=""
            depends=""
                msg=""
         jsFunction="org.apache.commons.validator.javascript.validateUtilities"/>
</global>
</form-validation>
```

web.xml

```
<display-name>SISACAD</display-name>
<description>SISTEMA DE ADMINISTRACION ACADEMICA (SISACAD)</description>
```

```
<filter>
  <filter-name>Security Chain</filter-name>
  <filter-class>com.bvg.comun.web.security.SecurityFilter</filter-class>
  <init-param>
    <param-name>loginAuthorization</param-name>
    <param-value>security</param-value>
  </init-param>
  <init-param>
     <param-name>guestUser</param-name>
     <param-value>INVITADO</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>Security Chain</filter-name>
     <url-pattern>/*</url-pattern>
</filter-mapping>

<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <load-on-startup>3</load-on-startup>
</servlet>

<servlet-mapping>
   <servlet-name>action</servlet-name>
   <url-pattern>*.do</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>5</session-timeout>
</session-config>
<mime-mapping>
   <extension>html</extension>
   <mime-type>text/html</mime-type>
</mime-mapping>
<mime-mapping>
   <extension>txt</extension>
   <mime-type>text/plain</mime-type>
</mime-mapping>
<jsp-config>
        <taglib>
            <taglib-uri>http://displaytag.sf.net</taglib-uri>
            <taglib-location>/WEB-INF/tld/displaytag.tld</taglib-location>
        </taglib>
        <taglib>
```

```
        <taglib-uri>http://jakarta.apache.org/struts/tags-html</taglib-uri>
        <taglib-location>/WEB-INF/tld/struts-html.tld</taglib-location>
      </taglib>
      <taglib>
        <taglib-uri>http://jakarta.apache.org/struts/tags-bean</taglib-uri>
        <taglib-location>/WEB-INF/tld/struts-bean.tld</taglib-location>
      </taglib>
      <taglib>
        <taglib-uri>http://java.sun.com/jsp/jstl/core</taglib-uri>
        <taglib-location>/WEB-INF/tld/c.tld</taglib-location>
      </taglib>
      <taglib>
        <taglib-uri>http://java.sun.com/jsp/jstl/fmt</taglib-uri>
        <taglib-location>/WEB-INF/tld/fmt.tld</taglib-location>
      </taglib>
      <taglib>
        <taglib-uri>http://jakarta.apache.org/taglibs/string-1.1</taglib-uri>
        <taglib-location>/WEB-INF/tld/taglibs-string.tld</taglib-location>
      </taglib>
  </jsp-config>

  <welcome-file-list>
      <welcome-file>/index.jsp</welcome-file>
  </welcome-file-list>

  <error-page>
     <error-code>404</error-code>
        <location>/web/jsp/error/404.jsp</location>
  </error-page>

  <error-page>
     <error-code>500</error-code>
          <location>/web/jsp/error/500.jsp</location>
  </error-page>
</web-app>
```

index.jsp

```
<%
System.out.print("simon");
response.sendRedirect(response.encodeRedirectURL("/"+request.getContextPath()+"/global.
do"));
System.out.print("simon");
%>
```

intro.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%>
<html>
<head>
  <%@ include file="/web/jsp/include/include.jsp" %>
  <script type="text/javascript">
```

```
        window.onresize = function() {
            sTitleArray = ['Admin'];
            for (var i=0; i<sTitleArray.length; i++){
                siFrameID = "iFrame" + sTitleArray[i];
                var f = document.getElementById(siFrameID);
                var h = "";
                if ( f.contentDocument ) {
                    h = f.contentDocument.documentElement.offsetHeight + 20 + "px";
                    f.setAttribute( "height", h );
                    f.parentNode.setAttribute( "height", h );
                } else if ( f.contentWindow ) {
                    h = f.contentWindow.document.body.scrollHeight + 5 + "px";
                    f.setAttribute( "height", h );
                    f.parentNode.setAttribute( "height", h );
                }
            }
        };
    </script>
</head>

<body bgcolor="#ffffff">
<table border="0" cellspacing="0" cellpadding="0" width="100%">
  <tr>
     <td width="182" valign="top" bgcolor="#00558C">
        <table width="182" border="0" cellspacing="0" cellpadding="0">
            <tr height="37px">
     <td  width="182">
        <div id="titulo_app"> <bean:message key="web.ui.titulo" /></div>
     </td> </tr>
     <tr height="10px"><td></td></tr>
     <tr>
        <td><img src="web/img/menu_sup.gif" width="182" height="17"></td>
     </tr>
       <tr><td bgcolor="#D5E2F1"><%@ include file="/side_menu.jsp" %></td></tr>
     <tr>
        <td><img src="web/img/menu_inf.gif" width="182" height="17"></td>
     </tr>
        <tr><td height="50px"/></tr>
     </table>
    </td>
    <td width="24" valign="top" background="web/img/index_11.gif">
   <table cellpadding="0" cellspacing="0" border="0" width="24">
   <tr><td><img src="web/img/index_10.gif" width="24" height="18"></td></tr>
   <tr><td><img src="web/img/index_3.gif" width="24" height="8"></td></tr>
   <tr><td><img src="web/img/index_7.gif" width="24" height="23"></td></tr>
</table>
</td>
  <td valign="top">
  <table cellpadding="0" cellspacing="0" border="0" width="100%">
  <tr>
     <td width="23" bgcolor="#00558C" valign="top"><img src="web/img/index_9.gif"
         width="23" height="18"></td>
     <td bgcolor="#00558C" valign="middle" align="right"><table cellpadding="0"
```

```
      cellspacing="0" border="0"><tr><td class="usuario_app_title"
      valign="middle">Usuario:</td><td class="usuario_app"><c:choose><c:when test="${not
      empty usuario.persona.nombresCompletos}">${usuario.gePersona.nombresCompletos}
      </c:when><c:otherwise>${usuario.id}</c:otherwise></c:choose>${usuario.persona
      .nombresCompletos}</td><td width="15px"/><td valign="bottom"><img
      onclick="location.href='logoff.jsp'" style="cursor:pointer;" src="web/img/cerrar.gif"
      width="14" height="14" /></td><td valign="middle" class="usuario_app"> <a
      href="logoff.jsp" >Salir</a></td></tr></table></td>
  </tr>
  <tr>
     <td colspan="2" valign="top" align="left"><img src="web/img/index_4.gif" width="23"
      height="8"></td>
  </tr>
  <tr>
  <td valign="top" style="vertical-align:top" colspan="2"> <iframe frameborder="0"
   marginwidth="0" id="iFrameAdmin" name="iFrameAdmin" scrolling="no"style="width:100%"
   src="${contextPath}/global.do? method=bienvenida" onload="resize(this.id)" ></iframe>
 </td>  </tr>   </table>  </td>   </tr>
</table>  </body>  </html>
```

login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="com.bvg.comun.web.security.SecurityFilter" %>
<%@ page import="com.bvg.comun.security.util.UsuarioLogin" %>
<%@ taglib uri="/WEB-INF/tld/c.tld" prefix="c" %>
<%
UsuarioLogin userlogin = (UsuarioLogin) request.getAttribute
(SecurityFilter.SESSION_USERLOGIN);
if (userlogin != null){
   request.getRequestDispatcher("/global.do").forward(request, response);
}
%>

<html>
<head>
</head>
<body>
  <h1>Autentificación de Ingreso</h1>
  <font color="black">
  <c:if test="${not empty SECURITY_MSGLOGIN}"> Tu login no fue exitoso. Intenta de
     nuevo!<br>
  Razón: <font color="red"><%= request.getAttribute (SecurityFilter.SECURITY_MSGLOGIN)
%>
</font> </c:if>
</font>
  <form action="<c:url value='j_security_check.html'/>" method="POST"onsubmit="u=
    document.forms[0].j_u; p=document.forms[0].j_p;
    if (u && u.value.length==0) {alert('Ingrese: Usuario');   u.focus(); return false; }
    if (p && p.value.length==0) {alert('Ingrese: Contraseña');p.focus(); return false; }
    return true;">
      <table align="center" width="60%">
```

```
<tr><td align="right">Usuario:</td><td><input maxlength="16" size="20" type='text'
   name='j_u'></td></tr>
<tr><td align="right">Contraseña:</td><td><input maxlength="16" size="20"
   type='password' name='j_p'></td></tr>
<%
    request.setAttribute("contextPath", request.getContextPath());
    request.setAttribute("sessid", request.getSession().getId());
%>
<tr><td colspan='2' align="right"><input value="Ingresar" name="submit"
  type="submit"><br><a href="#" onclick="if(document.all['j_u'].value) {location.href=
   '${contextPath}/security.do?method=olvidoClave&isOlvido=${sessid}&j_u=
   '+document.all['j_u'].value}else{alert('Ingrese el usuario');}">Olvido su clave?</a>
</td></tr> </table>
   </form>
   <script language="javascript">
         document.forms[0].j_u.focus();
   </script>
         <hr>
         <table style="width:100%"><tr>
                  <td><A href="<c:url value="/"/>">Inicio</A></td>
                  <td style="text-align:right;color:silver">ACADEMICO</td>
         </tr></table>
</body>
</html>
```

Mensajes

```
<%@ include file="/web/jsp/include/mensajes.jsp"%>
```

logoff.jsp

```
<%@ page import="javax.servlet.http.Cookie" %>
<%
session.invalidate();
Cookie terminate = new Cookie("Ultimo_User", null);
terminate.setMaxAge(0);
response.addCookie(terminate);

String cambio_de_clave = (String) request.getAttribute("cambio_de_clave");

if (cambio_de_clave != null && cambio_de_clave.compareTo("simon") == 0 ){
cambio_de_clave = "Su clave ha sido cambiada tiene que salir del sistema.";
}else{
response.sendRedirect(request.getContextPath()+"/");
}
%>
<%=cambio_de_clave%>
```

side_menu.jsp

```
<%@ include file="/web/jsp/include/taglibs.jsp"%>
<script type="text/javascript">
function cdd_menu647682(){//////////////////////////Start Menu Data/////////////////////////////////
```

```
//Unique Menu Id
        this.uid = 647682


/******************************************************************************************
                                                            Icon Images
******************************************************************************************/
    //Absolute positioned icon images (x,y positioned)

        this.abs_icon_image0 = "default_icon.gif"
        this.abs_icon_rollover0 = "default_icon.gif"
        this.abs_icon_image_wh0 = "7,7"
        this.abs_icon_image_xy0 = "-12,6"

        this.abs_icon_image1 = "main_divider.gif"
        this.abs_icon_rollover1 = "main_divider.gif"
        this.abs_icon_image_wh1 = "153,1"
        this.abs_icon_image_xy1 = "-155,2"

/******************************************************************************************
                            Global - Menu Container Settings
******************************************************************************************/
        this.menu_background_color = "#D5E2F1"
        this.menu_border_color = "#D5E2F1"
        this.menu_border_width = 1
        this.menu_padding = "2,2,2,2"
        this.menu_border_style = "solid"
        this.divider_caps = false
        this.divider_width = 0
        this.divider_height = 0
        this.divider_background_color = "#6F9CCE"
        this.divider_border_style = "none"
        this.divider_border_width = 0
        this.divider_border_color = "#D5E2F1"
        this.menu_is_horizontal = false
        this.menu_width = "100"
        this.menu_xy = "-14,-19"

/******************************************************************************************
                            Global - Menu Item Settings
******************************************************************************************/
        this.menu_items_background_color_roll = "#6F9CCE"
        this.menu_items_text_color = "#6F9CCE"
        this.menu_items_text_decoration = "none"
        this.menu_items_font_family = "Verdana"
        this.menu_items_font_size = "11px"
        this.menu_items_font_style = "normal"
        this.menu_items_font_weight = "normal"
        this.menu_items_text_align = "left"
        this.menu_items_padding = "2,5,2,7"
        this.menu_items_border_style = "solid"
        this.menu_items_border_color = "#6F9CCE"
        this.menu_items_border_width = "0"
```

```
        this.menu_items_width = 90
        this.menu_items_text_color_roll = "#000000"
        this.menu_items_border_color_roll = "#9b9999"

/*****************************************************************************************
                    Main Menu Settings
*****************************************************************************************/
      this.menu_background_color_main = "#D5E2F1"
      this.menu_items_background_color_roll_main = "#F5F5F5" //D5E2F1 // fondo
      sombreado del item seleccionado
      this.menu_items_text_color_main = "#333333"
      this.menu_items_text_color_roll_main = "#525252"
      this.menu_items_font_size_roll = "10px"
      this.menu_items_font_weight_roll = "bold"

      this.menu_items_border_color_roll_main = "# 999999" //borde externo degradado
      this.menu_padding_main = "2,2,2,2"
      this.menu_width_main = 182
      E8F1F7
      this.menu_is_horizontal_main = false
      this.divider_height_main = 0
      this.divider_background_color_main = "#999999"
      ${menu}

}///////////////////////// END Menu Data ////////////////////////////////////////

      </script>

<!--** START NAVSTUDIO MENU TAGS (647682) **-->
<div id="vqp_about" style="position:absolute;visibility:hidden;"></div>
  <script language="JavaScript" vqptag="doc_level_settings" is_vqp_html=1
    vqp_datafile0="side_menu.js" vqp_uid0=647682>cdd__codebase = "./web/js/menu/";
    cdd__codebase647682 = "./web/js/menu/";</script>
  <script language="JavaScript" vqptag="datafile" src="./web/js/menu/side_menu.js"></script>
  <script vqptag="placement" vqp_menuid="647682"
    language="JavaScript">create_menu(647682)</script>
<!--** END NAVSTUDIO MENU TAGS **-->
```